Class: Index

```java
import java.util.List;
import controlP5.*;

import org.json.JSONObject;

import codeanticode.glgraphics.GLConstants;
import de.fhpotsdam.unfolding.UnfoldingMap;
import de.fhpotsdam.unfolding.data.Feature;
import de.fhpotsdam.unfolding.data.GeoJSONReader;
import de.fhpotsdam.unfolding.geo.Location;
import de.fhpotsdam.unfolding.marker.Marker;
import de.fhpotsdam.unfolding.utils.MapUtils;
import processing.core.*;

public class Index extends PApplet {

	/**
	 * 
	 */
	private static final long serialVersionUID = 1L;

	ControlP5 cp5;

	PFont font;

	String climateData[], crimeData[], climateChunk[], crimeChunk[], popData[],
			popChunk[];

	Crime crimeArray[];
	Climate climateArray[];
	Population popArray[];

	UnfoldingMap map;
	List<Marker> stateMarkers;
	List<Feature> states;

	CheckBox crimeTypeBox, crimeBox, yearBox;

	public void setup() {
		size(1700, 1000, GLConstants.GLGRAPHICS);
		smooth();

		font = loadFont("Nilland-48.vlw");

		cp5 = new ControlP5(this);

		map = new UnfoldingMap(this, 0, 0, 1200, 1000);

		// Show map around the location in the given zoom level.
		map.zoomAndPanTo(new Location(38.951252f, -102.26932f), 4);
		map.setBackgroundColor(240);
```

```
// Add mouse and keyboard interactions
MapUtils.createDefaultEventDispatcher(this, map);

states = GeoJSONReader.loadData(this, "states-20m.json");
stateMarkers = MapUtils.createSimpleMarkers(states);
map.addMarkers(stateMarkers);

font = loadFont("Nilland-48.vlw");

climateData = loadStrings("climate_by_states.csv");
crimeData = loadStrings("CrimeStatebyState.csv");
popData = loadStrings("pop.csv");

climateArray = new Climate[climateData.length];
crimeArray = new Crime[crimeData.length - 1];
popArray = new Population[popData.length - 1];

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

    climateChunk = split(climateData[i], ",");
    climateArray[i] = new Climate(this, climateChunk[0],
            Float.valueOf(climateChunk[2]));

    // println("State: "+climateArray[i].getState());
    // println("Average temperature: "+climateArray[i].getTempC());
    // println();
}

for (int i = 1; i < crimeData.length; i++) {
    crimeChunk = split(crimeData[i], ",");
    crimeArray[i - 1] = new Crime(this, crimeChunk[0], crimeChunk[1],
            crimeChunk[2], Integer.parseInt(crimeChunk[3]),
            Integer.parseInt(crimeChunk[4]));
}

// for (int i = 0; i < crimeArray.length; i++) {
// println("State: " + crimeArray[i].getState());
// println("Type of Crime: " + crimeArray[i].getCrimeType());
// println("Crime: " + crimeArray[i].getCrime());
// println("Year: " + crimeArray[i].getYear());
// println("Count: " + crimeArray[i].getCount());
// println();
// }

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

    popChunk = split(popData[i], ",");
    popArray[i - 1] = new Population(this, popChunk[4],
            Integer.parseInt(popChunk[6]),
            Integer.parseInt(popChunk[7]),
            Integer.parseInt(popChunk[8]),
            Integer.parseInt(popChunk[9]),
            Integer.parseInt(popChunk[10]),
            Integer.parseInt(popChunk[11]));
```

```
        }

        // for (int i = 0; i < popArray.length; i++) {
        // println("State: " + popArray[i].getState());
        // println("00: " + popArray[i].getP0());
        // println("01: " + popArray[i].getP1());
        // println("02: " + popArray[i].getP2());
        // println("03: " + popArray[i].getP3());
        // println("04: " + popArray[i].getP4());
        // println("05: " + popArray[i].getP5());
        // println();
        // }

        crimeTypeBox = cp5.addCheckBox("Type of Crime", 1400, 90)
                .setSize(15,15)
                .addItem("Violent Crime", 0)
                .addItem("Property Crime", 1);

        crimeBox = cp5.addCheckBox("Crime", 1400, 140).setSize(15, 15);
        crimeBox.addItem("Murder and nonnegligent Manslaughter", 0);
        crimeBox.addItem("Forcible rape", 1);
        crimeBox.addItem("Robbery", 2);
        crimeBox.addItem("Aggravated assault", 3);
        crimeBox.addItem("Burglary", 4);
        crimeBox.addItem("Larceny-theft", 5);
        crimeBox.addItem("Motor vehicle theft", 6);

        yearBox = cp5.addCheckBox("Year", 1400, 290)
                .setSize(15, 15)
                .addItem("2000", 0)
                .addItem("2001", 1)
                .addItem("2002", 2)
                .addItem("2003", 3)
                .addItem("2004", 4)
                .addItem("2005", 5);
//       shadeStates();
    }

    public void draw() {
        background(50);
        writeText();
        map.draw();
    }

    public void shadeViolentStates() {
        int col = 0;

        for (Marker marker : stateMarkers) {
//           print(marker.getProperties().get("NAME").toString());
//           print(": ");
//           println(totalViolentCrime(marker.getProperties().get("NAME")
//           .toString()));

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

```
                          if (marker.getProperties().get("NAME").toString()
                                       .equalsIgnoreCase(crimeArray[i].getState()))
{

                                 if (totalViolentCrime(crimeArray[i].getState()) <=
822585.8) {
                                       col = 25;
                                 }
                                 if (totalViolentCrime(crimeArray[i].getState()) <=
822585.8 * 2
                                              &&
totalViolentCrime(crimeArray[i].getState()) > 822585.8) {
                                       col = 50;
                                 }
                                 if (totalViolentCrime(crimeArray[i].getState()) <=
822585.8 * 3
                                              &&
totalViolentCrime(crimeArray[i].getState()) > 822585.8 * 2) {
                                       col = 75;
                                 }
                                 if (totalViolentCrime(crimeArray[i].getState()) <=
822585.8 * 4
                                              &&
totalViolentCrime(crimeArray[i].getState()) > 822585.8 * 3) {
                                       col = 100;
                                 }
                                 if (totalViolentCrime(crimeArray[i].getState()) <=
822585.8 * 5
                                              &&
totalViolentCrime(crimeArray[i].getState()) > 822585.8 * 4) {
                                       col = 125;
                                 }
                                 if (totalViolentCrime(crimeArray[i].getState()) <=
822585.8 * 6
                                              &&
totalViolentCrime(crimeArray[i].getState()) > 822585.8 * 5) {
                                       col = 150;
                                 }
                                 if (totalViolentCrime(crimeArray[i].getState()) <=
822585.8 * 7
                                              &&
totalViolentCrime(crimeArray[i].getState()) > 822585.8 * 6) {
                                       col = 175;
                                 }
                                 if (totalViolentCrime(crimeArray[i].getState()) <=
822585.8 * 8
                                              &&
totalViolentCrime(crimeArray[i].getState()) > 822585.8 * 7) {
                                       col = 200;
                                 }
                                 if (totalViolentCrime(crimeArray[i].getState()) <=
822585.8 * 9
                                              &&
totalViolentCrime(crimeArray[i].getState()) > 822585.8 * 8) {
                                       col = 225;
                                 }
                                 if (totalViolentCrime(crimeArray[i].getState()) <=
822585.8 * 10
                                              &&
totalViolentCrime(crimeArray[i].getState()) > 822585.8 * 9) {
                                        col = 250;
                                 }

                                 marker.setColor(color(255, 0, 0, col));
                          }

                    }

             }

      }

      public void shadePropertyStates() {
             int col = 0;

             for (Marker marker : stateMarkers) {
//                   print(marker.getProperties().get("NAME").toString());
//                   print(": ");
//                   println(totalViolentCrime(marker.getProperties().get("NAME")
//                   .toString()));

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

                          if (marker.getProperties().get("NAME").toString()
                                        .equalsIgnoreCase(crimeArray[i].getState()))
{

                                 if (totalPropertyCrime(crimeArray[i].getState()) <=
822585.8) {

                                        col = 25;
                                 }
                                 if (totalPropertyCrime(crimeArray[i].getState()) <=
822585.8 * 2
                                              &&
totalPropertyCrime(crimeArray[i].getState()) > 822585.8) {
                                        col = 50;
                                 }
                                 if (totalPropertyCrime(crimeArray[i].getState()) <=
822585.8 * 3
                                              &&
totalPropertyCrime(crimeArray[i].getState()) > 822585.8 * 2) {
                                        col = 75;
                                 }
                                 if (totalPropertyCrime(crimeArray[i].getState()) <=
822585.8 * 4
                                              &&
totalPropertyCrime(crimeArray[i].getState()) > 822585.8 * 3) {
                                        col = 100;
                                 }
                                 if (totalViolentCrime(crimeArray[i].getState()) <=
822585.8 * 5
                                              &&
totalViolentCrime(crimeArray[i].getState()) > 822585.8 * 4) {
                                        col = 125;
                                 }
                                 if (totalPropertyCrime(crimeArray[i].getState()) <=
822585.8 * 6
                                              &&
totalPropertyCrime(crimeArray[i].getState()) > 822585.8 * 5) {
                                        col = 150;
                                 }
                                 if (totalPropertyCrime(crimeArray[i].getState()) <=
822585.8 * 7
                                              &&
totalPropertyCrime(crimeArray[i].getState()) > 822585.8 * 6) {
                                        col = 175;
                                 }
                                 if (totalPropertyCrime(crimeArray[i].getState()) <=
822585.8 * 8
                                              &&
totalPropertyCrime(crimeArray[i].getState()) > 822585.8 * 7) {
                                        col = 200;
                                 }
                                 if (totalPropertyCrime(crimeArray[i].getState()) <=
822585.8 * 9
                                              &&
totalPropertyCrime(crimeArray[i].getState()) > 822585.8 * 8) {
                                        col = 225;
                                 }
                                 if (totalPropertyCrime(crimeArray[i].getState()) <=
822585.8 * 10
                                              &&
totalPropertyCrime(crimeArray[i].getState()) > 822585.8 * 9) {
                                        col = 250;
                                 }

                                 marker.setColor(color(255, 0, 0, col));
                           }

                    }

             }

       }

       public int totalViolentCrime(String state) {
             int total = 0;

             for (int i = 0; i < crimeArray.length; i++) {
                    if (crimeArray[i].getYear() >= 2000 &&
crimeArray[i].getCrimeType().equalsIgnoreCase("Violent Crime")) {
                           if (crimeArray[i].getState().equalsIgnoreCase(state))
                                 total = total + crimeArray[i].getCount();
                    }
```

```
        }
        return total;
    }

    public int totalPropertyCrime(String state) {
        int total = 0;

        for (int i = 0; i < crimeArray.length; i++) {
            if (crimeArray[i].getYear() >= 2000 &&
crimeArray[i].getCrimeType().equalsIgnoreCase("Property Crime")) {
                if (crimeArray[i].getState().equalsIgnoreCase(state))
                    total = total + crimeArray[i].getCount();
            }
        }
        return total;
    }

    public void writeText() {
        fill(255);
        smooth();
        textFont(font, 32);
        textAlign(LEFT);
        text("Crime, Climate and Population", 1250, 50);

        textFont(font, 18);
        text("Type of Crime: ", 1250, 100);
        text("Crime: ", 1250, 150);
        text("Year: ", 1250, 300);
    }

    public void controlEvent(ControlEvent theEvent) {
        if (theEvent.isFrom(crimeTypeBox)) {
            if (crimeTypeBox.getArrayValue()[0] == 1) {
                shadeViolentStates();
            } else if (crimeTypeBox.getArrayValue()[1] == 1) {
                shadePropertyStates();
            }
        }
    }
}
```

Class: Crime

```
import processing.core.PApplet;
import processing.core.PConstants;
import processing.core.PFont;

public class Crime {

	String state, crimeType, crime;
	int year, count;

	PApplet parent;

	//Constructor
	Crime (PApplet p, String state, String crimeType, String crime, int year, int
count) {
		parent = p;

		this.state = state;
		this.crimeType = crimeType;
		this.crime = crime;
		this.year = year;
		this.count = count;
	}



	//Setters and Getters
	public String getState() {
		return state;
	}

	public String getCrimeType() {
		return crimeType;
	}

	public String getCrime() {
		return crime;
	}

	public int getYear() {
		return year;
	}

	public int getCount() {
		return count;
	}

	public void setState(String state) {
		this.state = state;
	}

	public void setCrimeType(String crimeType) {
		this.crimeType = crimeType;
```

```
    }

    public void setCrime(String crime) {
        this.crime = crime;
    }

    public void setYear(int year) {
        this.year = year;
    }

    public void setCount(int count) {
        this.count = count;
    }

}
```

Class: Climate

```
import processing.core.PApplet;

public class Climate {

    String state;
    float tempC;

    PApplet parent;

    //Constructor
    Climate (PApplet p, String state, float tempC) {
        parent = p;

        this.state = state;
        this.tempC = tempC;
    }


    //Setters and Getters
    public String getState() {
        return state;
    }

    public float getTempC() {
        return tempC;
    }

    public void setState(String state) {
        this.state = state;
    }

    public void setTempC(float tempC) {
        this.tempC = tempC;
```

```
	}}
```

Class: Population

```
import processing.core.PApplet;
import processing.core.PConstants;
import processing.core.PFont;

public class Population {

	String state;
	int p0, p1, p2, p3, p4, p5;

	PApplet parent;

	//Constructor
	Population (PApplet p, String state, int p0, int p1, int p2, int p3, int p4, 
int p5) {
		parent = p;

		this.state = state;
		this.p0 = p0;
		this.p1 = p1;
		this.p2 = p2;
		this.p3 = p3;
		this.p4 = p4;
		this.p5 = p5;

	}

	//Getters
	public String getState() {
		return state;
	}
	public int getP0() {
		return p0;
	}
	public int getP1() {
		return p1;
	}
	public int getP2() {
		return p2;
	}
	public int getP3() {
		return p3;
	}
	public int getP4() {
		return p4;
	}
	public int getP5() {
		return p5;
	}
}
```