Leila Maritim and Muhammad Bilal

GoogleEarthTweetMap

This code is a main method for a Google Earth tweet mapper program.

It imports various classes and libraries such as `BufferedImage`, `IOException`, `ParseException`, `ArrayList`, `CsvReader`, `KmlWriter`, `WmsConnector`, and `ImageOverlay`.

The program reads data from a CSV file and creates a KML file with it using `KmlWriter`.

It then retrieves a WMS image using the `WmsConnector` class and creates an image overlay on top of the KML file using the `ImageOverlay` class.

Finally, the program runs an external program, Google Earth, passing the file paths of both KML files as parameters.

WmsConnector

This code defines a class named `"WmsConnector"`, which retrieves and returns a Web Map Service (WMS) image as a `BufferedImage` object.

The class has a single static method named `"retrieveImage"` which accepts a URL string as input and returns the retrieved image as a `BufferedImage` object.

The method creates a `WebMapServer` object using the input URL string and then creates a `GetMapRequest` object using the `WebMapServer` object.

The `GetMapRequest` object is configured with parameters such as version, transparency, bounding box, dimensions, layer, format, and spatial reference system.

The method then issues the `GetMapRequest` to retrieve the WMS image.

If the response content type is "image/png", the method reads the response input stream into a `BufferedImage` object and writes the image to a file.

If the response content type is not "image/png", the method prints an error message. Finally, the method returns the retrieved WMS image as a `BufferedImage` object.

CsvReader

This code defines a class `CsvReader`

The class contains a method `readCSV` that reads a CSV file and stores its content in an `ArrayList` of `String` arrays.

The method takes a `String` argument `file` which specifies the path of the CSV file to read.

The method begins by creating an empty `ArrayList` to store the CSV data.

It then declares a variable `line` to store each line of the CSV file and a boolean variable `firstLine` to indicate whether it is the first line (which is typically the header).

It then opens the CSV file using a `BufferedReader` and reads each line of the CSV file using the `readLine` method.

`If firstLine is true, the code skips the first line (header) by setting firstLine to` false and using the `continue` statement to jump to the next iteration of the loop. If it is not the first line, the method splits the line into an array of `String`s using the delimiter `;` and adds the resulting array to the `ArrayList`.

If an `IOException` occurs, the method prints the stack trace using the `printStackTrace` method. Finally, the method returns the `ArrayList` of CSV data.

KmlWriter

The class has a private instance variable “data” of type `ArrayList<String[]>` to hold the input data for writing KML.

The constructor initializes the “data” variable with input data.

The private method: `"timestampToExtrude"` converts a timestamp to a boolean value for the extrude tag, while the public method and `"writeKML"` writes a KML file given an `ArrayList` of String arrays and a file path.

The `"writeKML"` method opens a file to write to, defines the beginning of the KML file, including namespace and Document tag, and adds Placemark tags with corresponding data to the KML file by iterating over the data.

The latitude, longitude, and date are parsed from the input data, and the output date is formatted according to the KML standard(`"yyyy-MM-dd'T'HH:mm:ss'Z'"`).

The method also defines a style for Placemark icons and includes additional data in ExtendedData tags.

Finally, the Document and kml tags are closed, and the KML string is written to the file and the writer is closed. If an exception occurs during the writing process, the stack trace is printed.

ImageOverlay

The code defines a class called "`ImageOverlay`" which has fields for a name, an image URL, and north, south, east, and west coordinates.

The constructor takes the image URL as a parameter and sets the corresponding field into the href element in the KML.

The class also has a method called "`writeKMLOverlay`" that takes a file path and uses a `FileWriter` to write a KML file at that location.

The method first defines the beginning of the KML file, including the namespace and Document tag, and then appends the name, image URL, and coordinates to the `GroundOverlay` tag.

Finally, it closes the Folder and kml tags and writes the resulting KML string to the file. If an IOException occurs, it prints the stack trace.

Sample output of the program

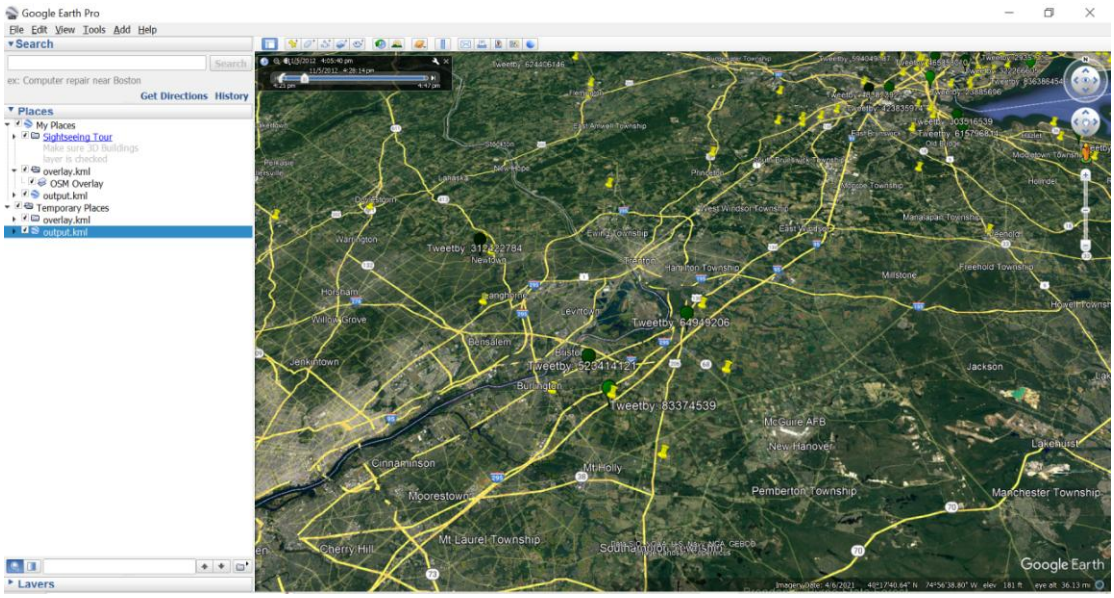

Fig 1. Sample view of tweets on Google Earth.

References

1. Google KML documentation:

https://developers.google.com/kml/documentation/kmlreference

2. Geotools WMS documentation:

https://docs.geotools.org/latest/javadocs/org/geotools/ows/wms/WMSRequest.html

3. Geootols WMS user guide:

https://docs.geotools.org/latest/userguide/index.html

4. Oracle BufferedImage class documenation

https://docs.oracle.com/javase/7/docs/api/java/awt/image/BufferedImage.html

5. Buffered Reader class documentation:

https://docs.oracle.com/javase/8/docs/api/java/io/BufferedReader.html

6. Documentation comments for Java programs guide:

https://www.oracle.com/technical-resources/articles/java/javadoc-tool.html#styleguide

7. https://stackoverflow.com/questions/22563986/understanding-getinputstream-and-getoutputstream