Google Earth Tweet Mapper

Introduction and Overview

This documentation provides a detailed account of the design and implementation decisions of the end-of-term assignment, which requires the creation of an executable Java class called *GoogleEarthTweetMapper* that integrates data from multiple sources and displays it in Google Earth. The data to be displayed comprises the content and attributes of tweets along a time series as well as an OGC WMS (Web Map Service) image to be integrated as a ground overlay in Google Earth.
Upon execution, the program completes the following tasks:

1. Downloads and parses a CSV file containing tweets and metadata
2. Connects to a WMS and downloads an image showing OpenStreetMaps
3. Writes the image and selected data from the CSV file into a KML (Keyhole Markup Language) file
4. Opens the KML file with Google Earth

The program uses GeoTools (modular open-source development toolkit used to implement the OGC specifications to create a connection to a WMS) and OpenCSV (lightweight open-source library used to read and parse the CSV file) as external libraries.

Design Decisions

An essential criterion for the application's design is a modular structure of the code, for which Java is a well-suited programming language. To achieve a general modularity, the project is organized into six classes (Figure 1), each with a distinct purpose. Among these classes, five are non-executable: *WMSConnector*, *CSVParser*, *FileDownloader*, *GoogleEarthLauncher*, and *KMLWriter*. The only class containing a *main()* method is *GoogleEarthTweetMapper*, which is responsible for executing the program by calling methods from the other classes. Furthermore, all of the classes use static methods to avoid needing to instantiate many objects. The Program also benefits from the use of external libraries, as they provide modular and preformant solutions and avoid the creation of redundant code.

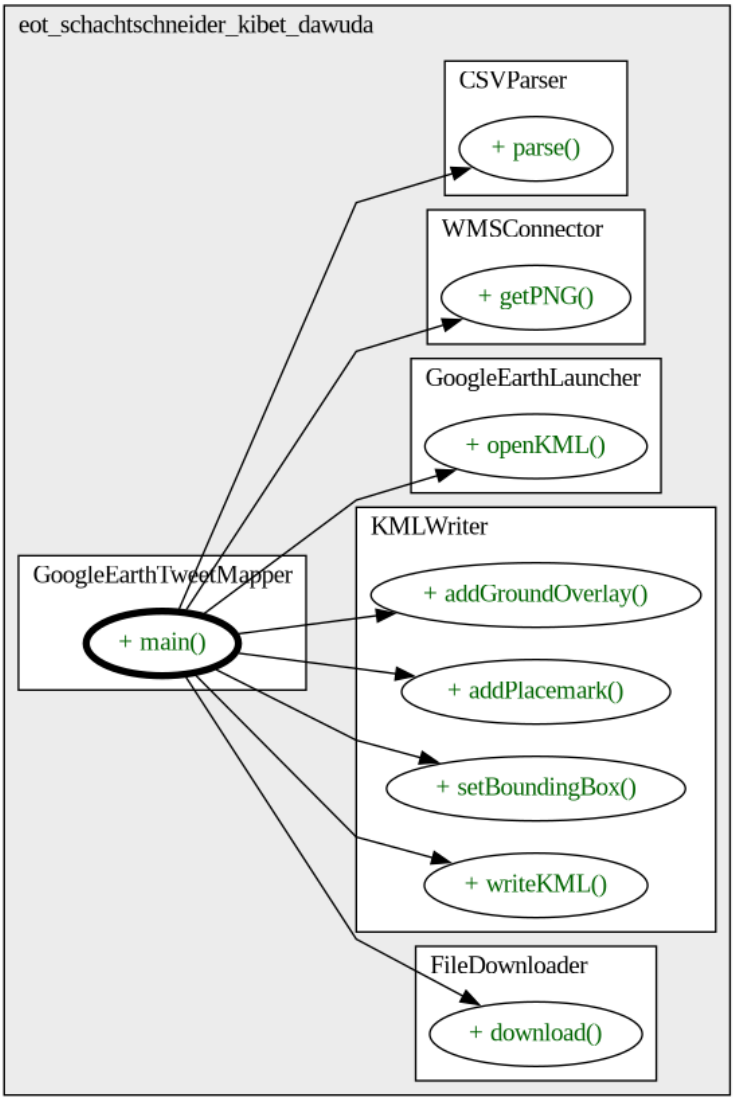

Figure 1: Call multigraph of the program

Implementation

The following section describes the implementation of each of the six classes used in this program.

Google Earth Tweet Mapper

A global *String* variable is defined to store the location of the KML file to be written in the working directory of the program. The *main()* method operates and utilizes the project's classes as follows:

- The *FileDownloader* class is used by calling the *download()* method to download the CSV file from *http://www.berndresch.com/work/twitter.csv*. The file is then stored in a *String* variable named *twitterCSV*.
- The *CSVParser* class is called to parse the contents of *twitterCSV* and extract the following information: latitude, longitude, tweet content, userID, and creation date and time.
- The *WMSConnector* class is used by calling the *getPNG()* method with the URL of the WMS and bounding box coordinates as arguments.
- The *KMLWriter* class is used to write the desired content of the CSV file and WMS image into a KML file. The file content is created by calling the class variables for the beginning and end of the KML file and the methods *addGroundOverlay()* (adds the ground overlay) and *addPlacemark()* (adds the tweet content and metadata). The use of chaining allows for the generation of the KML content in a single line. This class is also used to create and store the bounding box (also used by the *WMSConnector* class).
- The *GoogleEarthLauncher* class is used to launch Google Earth and display the contents of the KML file on a map. This is done by calling the method *openKML()* with the KML file as an argument.

This class is also responsible for the majority of exception handling of the program. This is done by utilizing try-catch blocks.

File Downloader

The public *FileDownloader* class contains the method *download()*, which the program calls to download the *tweets.csv* file. This removes the need to manually download the correct file and place it in the right directory to run the program successfully. The method takes the URL of the file to be downloaded and the name to be given to the file as parameters. A *URL* and *FileOutputStream* object are instantiated. The *URL* object is used by the

FileOutputStream object to open a stream and download the file. At the end of the method, the stream is closed.

CSV Parser

The *CSVParser* class is used to go through the CSV file containing the tweets line by line, extract certain values, and store them in variables, which can then be used as input for creating the KML file. The class uses the OpenCSV library to perform these operations. The class contains five ArrayList class variables that store the coordinates, tweet content, time of creation and the user ID of the author. These are implemented as *ArrayLists* as the *ArrayList* class allows for flexible expansion of the list, as opposed to *String* arrays. The public method *parse()* takes the location of the CSV to be parsed as a parameter and is used to perform the reading and writing tasks. In this method, a *FileReader* and a *CSVReader* object are instantiated. The *CSVReader* builds upon the *FileReaders* functionality to read and parse the CSV file. The type of separator used in the CSV file is also declared when instantiating the *CSVReader* object. Because the method traverses through the CSV file line by line, a *String* array is used to store the content of the current line being read. A *while* loop is responsible for reading the file line by line and adding the desired attributes to the respective *ArrayLists* until the file is fully read. The first line of the file is skipped as it only contains the attribute types. An *IOExeption* and *CsvValidationException* are caught.

WMS Connector

The *WMSConnector* class imports all of the necessary GeoTools packages for creating a WMS connection. A method called *getPNG()* is used to establish the connection and fetch the desired image. The method takes two parameters: *wmsURL* and *boundingBox*. *wmsURL* should contain a link to the map server (*https://maps.heigit.org/osm-wms/service?SERVICE=WMS&REQUEST=GetMap&VERSION=1.1.0*). The *boundingBox* defines the extent of the generated image using predefined coordinates stored as a single *String*. To return the final image, a *GetMapRequest* object is instantiated with the following specifications: image format: PNG, image dimension: 2000x2000 pixels, transparency: True, spatial reference system: EPSG:4326, and bounding box: -74.8, 40.18, -73.40, 41.10. Finally, if a PNG image is receieved, the method writes the image into the program's working directory on the hard drive under the name *wmsFile.png*.

KML Writer

This class is designed to write KML files based on tweet data and imagery provided to it. The KML format is used to display geographical data in Google Earth. The class has a class variable called *kmlBeginning* of type *String* which contains the beginning of a KML file, and *kmlEnd* which is also a *String* variable containing the end of a KML file. This allows for the modular creation of KML files, as the user can define the content and the order of the content placed in between these two elements. The class also contains a static *HashMap* object called *boundingBox*, which is used to store the extent of the bounding box (determines a geographic area that can be used for the ground overlay element).

The method *setBoundingBox()* is used to update the values of the *boundingBox* variable.

The method *writeKML()* is used to write the content to the KML file itself. This method takes two parameters - the name of the file to be written and the content of the KML file as a *String*. The method uses a *FileWriter* object to write the content to the file.

The method *addPlacemark()* is used to create a KML placemark element for each tweet. It takes in four *ArrayLists* as parameters - *lng*, *lat*, *tweet*, and *createdAt*, which contain the longitude, latitude, content of the tweet, and creation time of each tweet, respectively. The method loops through the *ArrayLists* and creates a placemark element for each tweet using the data provided. The placemarks are added to a *String* variable called *placemark* which is then returned.

The method *addGroundOverlay()* is used to create a KML ground overlay element. This method takes a parameter called *overlayImage* which is the location of the image file to be used as the overlay. The method returns a *String* variable called *groundOverlay* which contains the necessary KML tags to create the ground overlay element.

Google Earth Launcher

This class provides the functionality to open a KML file with Google Earth. The *setGoogleEarthPath()* method can be used to set the path to the Google Earth executable, allowing the user to specify the path for their system. The *openKML()* method takes the location of the KML file to open as an input parameter and uses the *Runtime.getRuntime().exec()* method to execute the command to open the file with Google Earth. The method also prints a message to the console to indicate that Google Earth is being launched.