



Basics of Software Development

Visualising Tweets in Google Earth

Dynamic Digital Mapping

Bernd Resch

End-of-term Assignment

End of Term Assignment

- Create an executable Java class called "*GoogleEarthTweetMapper*", which integrates data from the following sources and displays them in Google Earth:
 - ◆ OGC WMS
 - ◆ Tweets text file

End of Term Assignment

- **WMS Connection:** <http://129.206.228.72/cached/osm>
 - ➔ *Layer: "osm_auto:all"*
 - ◆ *Hint: to set up the WMS request, append the HTTP GET parameters to the URL above (...WMSServer?request=...)*
 - *WGS84 BBOX: -71.13,42.32,-71.03,42.42*
 - *Request the image in PNG format with transparent background*
 - ➔ *Takes a while to load! (try it in your web browser first)*

End of Term Assignment

- **Integration of the WMS Image into Google Earth:**
 - ◆ Create a Java programme that stores the WMS image as a file on your PC and integrate the local file into your KML structure (even though this is not the most efficient way to do this, it is a good programming exercise)
 - ◆ For integrating the WMS image into the KML structure, use a *GroundOverlay* element
 - 30% opacity (70% transparency)
<http://www.binaryhexconverter.com/hex-to-decimal-converter>

End of Term Assignment

- *Alternative WMSs*

(in case the other one doesn't work properly)

- ♦ <http://giswebservices.massgis.state.ma.us/geoserver/wms> → Layer:
"GISDATA.PARCEL_STATUS"
- ♦ <http://giswebservices.massgis.state.ma.us/geoserver/wms> → Layer:
"GISDATA.CENSUS2000TRACTS_POLY"

End of Term Assignment

- **Tweets text file:**

<http://www.berndresch.com/work/twitter.csv>

- ♦ “*lng*” and “*lat*” columns are the coordinates
- ♦ CRS: WGS84 → no reprojection necessary
- ♦ Visualise the columns “*tweet*” and “*created_at*” (e.g, in the <description> tag)
- ♦ *Hint: Parse the text file line by line, extract the values and then insert them into the KML structure.*

End of Term Assignment

■ Hints:

- ◆ Use the jar files (and code) used in previous lab exercises
- ◆ Make extensive use of object-oriented programming paradigms
➔ one Java class for each component (modularity!)
- ◆ For integrating the WMS image into the KML structure, use a *GroundOverlay* element
- ◆ Extracting the values from the Twitter CSV file will require String handling operations
<https://docs.oracle.com/javase/tutorial/java/data/manipstrings.html>

End of Term Assignment

- ***ADDENDUM (not mandatory):***
 - ◆ Visualise the tweets as a time series in Google Earth
 - Enable time series visualisation in Google Earth
<https://developers.google.com/kml/documentation/kmlreference#timestamp>
 - Display one of the attributes (e.g., time, user_id) visually, e.g. through
 - Extruding an attribute value
<https://developers.google.com/kml/documentation/kmlreference#polygon>
 - Colouring the extruded polygons according to their value
<https://developers.google.com/kml/documentation/kmlreference#style>
<http://colorbrewer2.org>
 - ...

End-of-term Assignment ::: Grading

- **Grading criteria**

- ◆ Degree of modularity, structure of the application
- ◆ Clarity, comprehensiveness, rationale and completeness of the documentation
 - Report (in English language)
 - Inline documentation

End-of-term Assignment ::: Formalities

- Write a short documentation
 - ◆ 4-5 pages incl. figures + references
 - ◆ Contents
 - Design decisions
 - Implementation details
- Put all the Java files in a package named "eot_<name1>_<name2>_<name3>"

End-of-term Assignment ::: Organisational

- Deadline: 22 February 2019
- Groups of 3 students
- Submission via email (bernd.resch@sbg.ac.at):
 - email subject "SWE End-of-term Assignment"
 - ◆ Code (.java files) → no .jar files!
 - ◆ Documentation

End-of-term Assignment ::: Organisational

- Final claim:
 - ◆ Please don't cheat!





Basics of Software Development

Visualising Tweets in Google Earth

Dynamic Digital Mapping

Bernd Resch