

GeoMashup Code Camp
University of Helsinki
Department of Computer Science

Jukka Leino
ID# 012607475
Otto Waltari
ID# 013182892
Md. Nazmul Haque Khan
ID# 014042177

Date: 31 October, 2011

Introduction

Nowadays different type of online map services has become very popular. Different platforms like Google, Bing and Ovi maps have made this easier to make application using spatial data and maps. Different spatial data providers and their api's made the data available to do very complex task in easy way. In our project we have used these api's and service to build a geo-location based application. Our software uses to twitter and Google map to show tweets from different person from a specific place and shows in the map with traffic congestion of that place at that moment which gives us an idea, how people uses twitter while they are in the traffic.

Features and Limitations

The initial idea of our application was to make an application with twitter, facebook api's and traffic congestion data and visualize in the Google map how many people use social media during time they are stuck in the traffic. In order to do that we have searched for suitable api from different social media like facebook, twitter, Google plus and foursquare but except twitter we did not get suitable api for our application. Finally in our application we have used twitter api along with Google map's traffic congestion data. Our application can search tweets from different person in a specific location and point the places from where they have tweeted. In the back ground there are traffic congestion layer over which our application plots the location of the tweets and thus we can get an impression about how many users are using twitter in the traffic jam. The other feature of our application is that we can search tweets according to specific query. We can also search tweets from a specific user by his twitter id and track his or her movement on the map by drawing lines between the locations of his or her tweets. Thus we can have an idea about a person's daily rout those who use twitter regularly.

Due to lack of api from other social networking websites we could not visualize the data from other social networking website. Our application has only fetches tweets with a radius of 8 km from the focus area because of the limitation of the api. Our application only shows the tweets in the map from those data which contain their geolocation data. Our application also has the limitation of fetching all the tweets of the location. It can only fetch tweets with a specific query. This is because in the search api of tweeter query field is a mandatory field and we can not search any tweets with this field.

User Guide

In our application there are eight buttons and one query text box. In our application the default location is set to “New York” and default query is set to “and”. So in the start of our application our application loads the tweets from New York and the tweets which includes “and”. Then if the user clicks show marker button in the map the user can see from where the tweets are coming from. By clicking on the marker the user can also see the tweets and see when it was tweeted. If the user wants look at older tweets then he or she can click the button older tweets and then show marker button that will instantly visualize the tweets on the map. Our application also keeps the option to look at the tweets at the bottom of the page and gives an option to see the individual tweet in the map by clicking show marker link beside the tweet. Figure 1 shows the screen shot of our applications default settings.

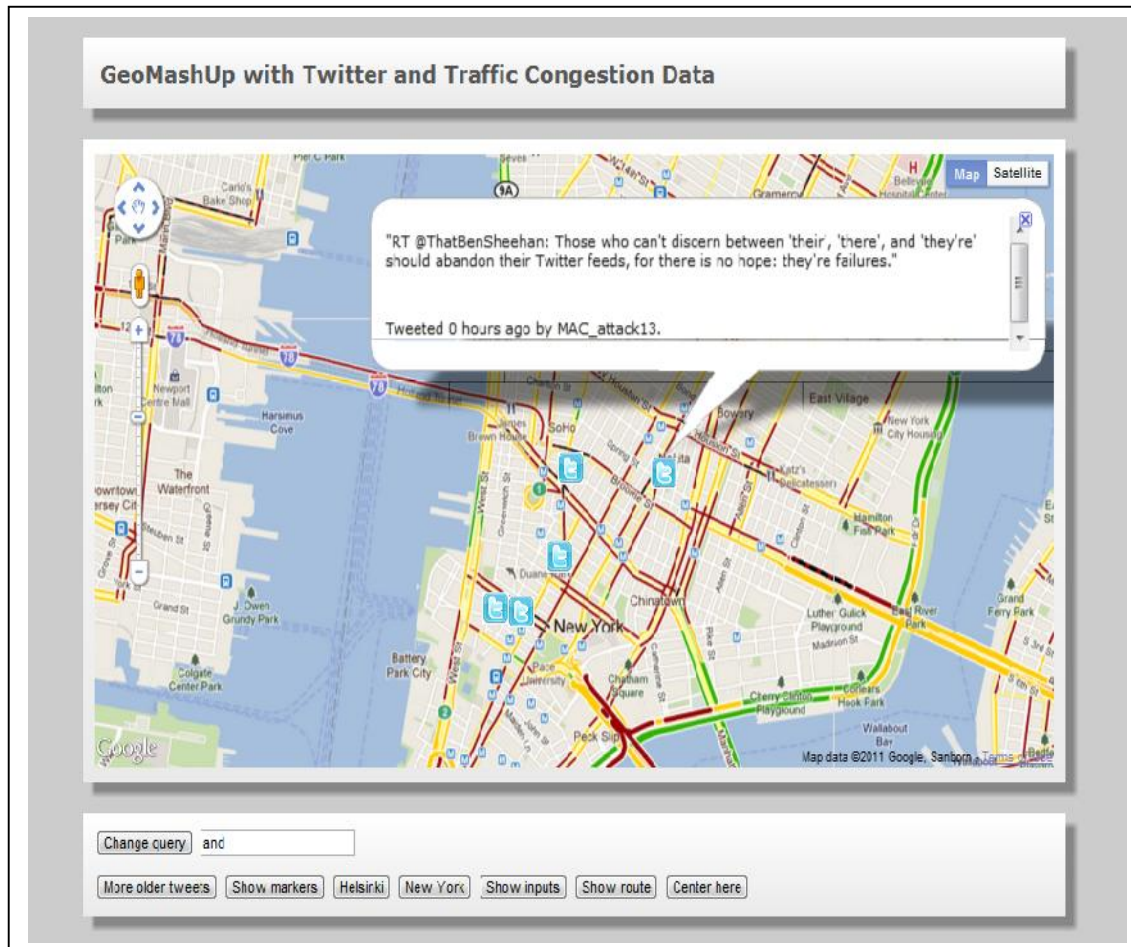


Figure1: Showing tweets from New York over Traffic layer

The user can change the query string by writing the query in the text box given beside the change query button and then clicking the change query button will fetch the tweets with change query. The user can set the focus area to either New York City or Helsinki by clicking the corresponding buttons. The user can also move the image by dragging mouse pointer and then select a place of his or her interest by clicking Centre here button to set the centre of the map.

In order to track a specific tweeter user the user needs set the query string with the prefix "from:". Then the user needs to click Show markers and then Show

route button. This action will draw lines between the tweets made by the user. Figure 2 shows the tweets from twitter user killer62 and tracks his route.

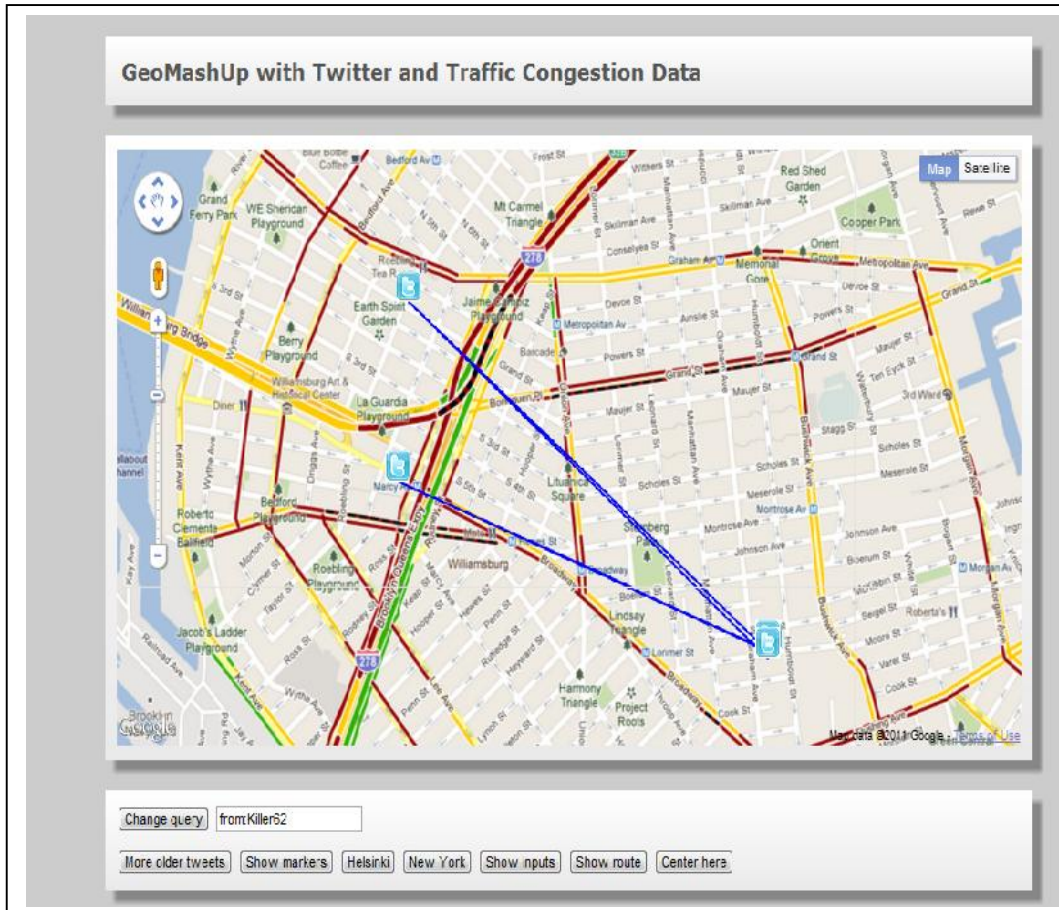


Figure2: Tracking the route of user killer62

Implementation

Google maps JavaScript API V3 was very clear to understand and to implement to our application. We used several overlays; one was the marker overlay other the polyline overlay and road congestion. All had nicely documented API functionality. Some problems were encountered due to the asynchronous nature of JavaScript, but with some debugging the issues were solved. JQuery JSON callback function was used to get the json data back from twitter. This data was then parsed to a 'tweet'-list which contained all the used

info about a tweet. Markers and polylines were drawn to the map using this data.

Because of the tight schedule and intensive nature of the course there wasn't much time for planning the project. We started the project by digging Google Maps and Twitter api's and wrote code as we learned how to use them. Everyone in our group worked mostly independently and we did not use any revision control, which made the cocktail even more interesting. The sketchy code was eventually wrapped together and tuned into working condition. The final project seems to work like a charm although the implementation is fairly inelegant.

Platforms and Tools

We have used plain HTML, JavaScript and CSS to develop our application. We have used Google map api to plot the spatial data and show traffic congestion. We have used twitter api to show tweet data. We have tested our application both in Google chrome and Firefox Mozilla web browser and works pretty nice. Our application was not in a very big scale so we have chosen to use plain JavaScript to develop our application. Working with plain JavaScript was a little bit problematic because it was quite difficult to debug. We also found little bit difficulty with json callback function. But overall this code camp was total learning experience and fun.