This repository Search Explore Gist Blog Help PY ppyordanov

ppyordanov / Dynamic-Noise-and-Pollution-Map PRIVATE

Unwatch 2 Star 0 Fork 0

Optimize Map Working Space Usage #45

Edit New issue

Closed **ppyordanov** opened this issue on Nov 10, 2014 · 2 comments

PY **ppyordanov** commented on Nov 10, 2014

At the moment, when in default zoom, the border frame that is used by the mobile web application to display sensor readings is getting over-populated with information after about a thousand samples have been added to the working space area. This will be optimized and there are a couple different techniques that can be used for the implementation.

PY **ppyordanov** self-assigned this on Nov 10, 2014

PY **ppyordanov** added this to the **Mobile Web Application** milestone on Nov 10, 2014

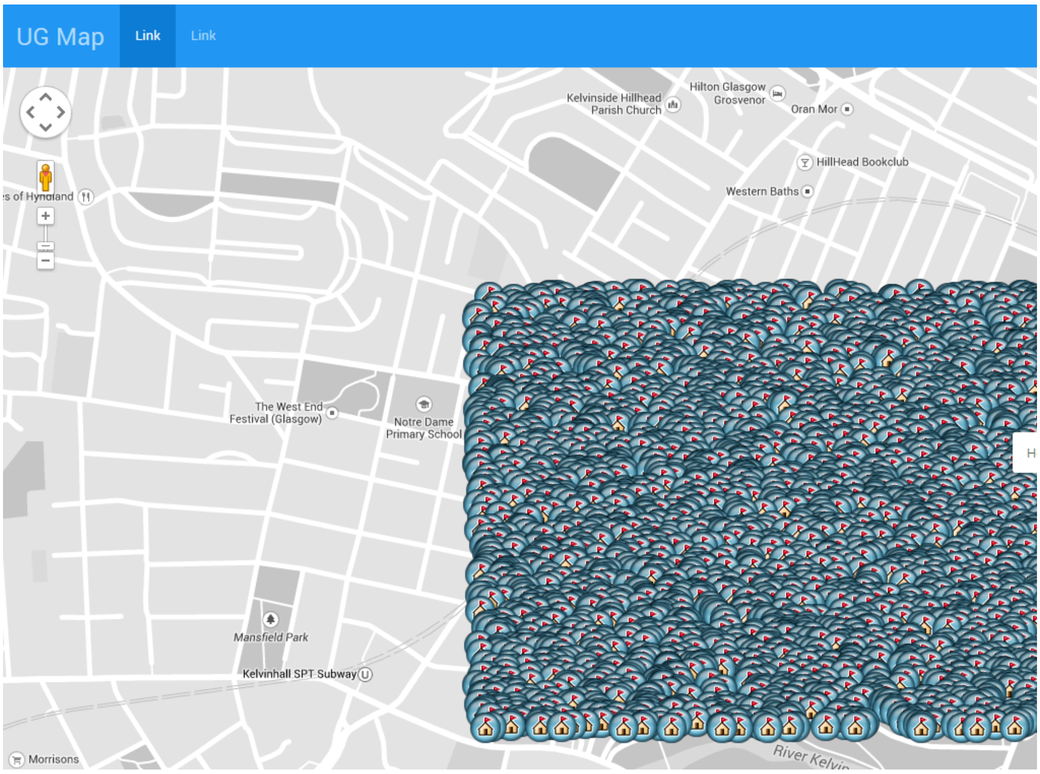
PY **ppyordanov** added the **enhancement** label on Nov 10, 2014

PY **ppyordanov** modified the milestone: **Additional functionality**, **Mobile Web Application** on Nov 10, 2014

PY **ppyordanov** commented on Nov 11, 2014

The maximum expected load is about 50 to 60 000 data readings from the Smart Citizen Kit. This might cause some map rendering issues and data filters should be efficiently implemented to ensure high performance and fast processing.

Example data visualization for 5 000 data readings without any optimizations applied (as seen on the screenshot location (latitude and longitude) is randomly generated within the specified aerial bounds to simulate data received from Android devices):



To tackle the large data set visualization, some of the following approaches can be used:

Labels

enhancement

Milestone

Mobile Web Applic...

Assignee

PY ppyordanov

Notifications

Unsubscribe

You're receiving notifications because you modified the open/close state.

1 participant

PY

Lock issue

- data aggregation - this option will involve multiple readings being aggregated into a single one, which is then stored in the database or store all the information in the relevant collection and then aggregate it at runtime (this will be more resource costly as it will involve more processing during runtime);
    - there are a number of methods that can be used for efficient data aggregation:
        - one would be to take the first out of each, say, 5 data readings as well as the 5th one (or arbitrarily choose a single one) and find their average - > this will help to reduce the normal inaccuracies when reading data /explained numerous times by the SCK team/, especially when 'the sensors go crazy'; for example, some big data outliers will inevitably be noticed as the kit is being carried around campus as it is going to swing back and forth constantly, which will be a side effect of the evaluation process as we are only interested in the actual data from the surrounding environment.
        - the other, more conservative approach, would involve grouping each , say 3/5 sensor data readings and finding their average (dr1 + dr2 + dr3 + dr4 + dr5)/5 which is going to be more accurate than the other technique in some cases, but will not help much in the elimination of data outliers as a single very high value might result in a significant change in the final result.

NOTE: considering each 5 data readings are grouped with a 30 sec. window, the system is going to be deriving a single data reading aggregation each 2,5 min. which, for the Glasgow University campus area, I would consider normal, taking into account walking speed; regarding location, the carrier client device's GPS system can, for example be used to store geolocation every 1,25 min so that from the 3 generated values for a period of 2,5 min, the middle one can be used. /these are just implementation examples/ideas and they will be changed as the system is developed further/

- change the size of the icon used for data visualization and experiment with default map zoom to enlarge the map working space as much as possible
- changing the data post window (increase to reduce number of sensor readings) -> least desirable, as we want more granularity and data accuracy

The notes here will provide food for thought so that the most efficient solution mechanism can be applied in the final implementation.

ppyordanov referenced this issue from a commit on Feb 4

`[LOGO] + [REFACTORING]` d01e8a9

ppyordanov referenced this issue from a commit on Feb 4

`[LARGE_ITERATION] improved models, better UI, map configuration` 32b208b

ppyordanov referenced this issue from a commit on Feb 4

`[REFACTORING] menu implementation and optimizations` 9d3b885

ppyordanov referenced this issue from a commit on Feb 4

`[ITERATION] map configuration` c50966c

ppyordanov referenced this issue from a commit on Feb 4

`[ITERATION_ROUTES] route data` 076432b

ppyordanov referenced this issue from a commit on Feb 4

`[IMPROVEMENTS] configuration + user feedback UI` 227138a

PY

ppyordanov commented 28 days ago

This task has been completed and can be closed now.

ppyordanov closed this 28 days ago

Write Preview

Markdown supported Edit in fullscreen

Leave a comment

Attach images by dragging & dropping, selecting them, or pasting from the clipboard.

Reopen issue Comment