

Application for summer of Code 2008: Shashank Singh

Synopsis:

The intention is to work on the idea suggested by GSoC's website with few inputs of my own; the primary objective that needs to be fulfilled for completion of program would be;

Namely,

- A) Creation of layers for displaying *Panoramio* geo-tagged photographs.
 - Creation of extension for *Marble* file format to include support for geo-tagged photographs and thumbnails.
- B) Implementation of parser for *JSON* or *JSONP*, which would be used to parse replies from REST API's of *Panoramio*.
- C) Creation of layers for displaying *Wikipedia* geo-tagged articles.
 - Creation of extension for *Marble* data file format to include support for *Wikipedia* articles and their respective thumbnail.
- D) Modification to current data downloading framework to download pictures and Wiki-articles.
- E) Modification to *Marble* client code to correctly draw the *Panoramio* / Wiki layer data supplied to it in our custom data file format.
- F) Integration with Digikam.(Optional)

All these ideas work toward making "*Marble*" a more main stream application with support for all the publicly available data, leading to better user experience.

Project Description:

The project can be divided into five major tasks;

Namely,

- A) Creation of layer for displaying *Panoramio* geo-tagged photographs, *Panoramio* uses REST API and returns the result in *JSON* or *JSONP*.

A sample REST query for *Panoramio* can be constructed like this:

```
http://www.panoramio.com/map/get_panoramas.php?order=popularity&set=public&from=0&to=20&minx=-180&miny=-90&maxx=180&maxy=90&size=medium
```

The usage limit is 100 photos per query. This usage limitation forces us to optimize our algorithm to decide on the fly which data needs to be handled first. Using progressive querying, for every zoom level the program need to intelligently decide which place mark to download.

Using cache we could decrease the need to query *Panoramio* servers.

- B) Implementation of parser for *JSON* or *JSONP* would be required to handle the output generated by *Panorama*'s API. *JSON* stands for JavaScript Object Notation and *JSONP* stands for *JSON* with Padding, *JSON* would require a special parser to be built around it using Qt Classes which could support two structures, namely collection and ordered pairs that *JSON* is built around.

Possible solutions include using QtScript to create Javascript objects from *JSON* and convert them to native Qt/C++ Data structure so that they could be used system wide. Another Solution could be creating a C/C++ parser for *JSON* from scratch and then exporting resultant Javascript objects to native C++ equivalents.

- C) Creation of layer for storing *Wikipedia* geo-tagged information in Marble's custom file format. The entire geo-referenced articles are available from *Wikipedia* World; a dump of geo-referenced article can be downloaded in SQL (or CSV) format, the database dump supplies geo-referenced articles in following languages
English ,German ,Spanish, French ,Italian ,Japanese Dutch ,Polish , Portuguese , Russia , Swedish ,Turkish ,Chinese , Catalan, Icelandic
The Database can be queried for latitude, longitude and country .Each Language would have its own layer, a suitable option would be added to GUI so as to facilitate users to select the language they want their articles to be in.
At any zoom level, the central database would be queried for place marks for articles; the query parameters would contain the longitude and latitude extremes of visible Quadtiles.
For example:
*Select count * from wpgeo
Where lat > x & lat < x1
And lon > y & lon < y1*
- D) Modification of the current map downloading framework to include downloading of *Panoramio* / *Wikipedia* geo-tagged data in our custom format.
This feature could be added by employing existing frameworks namely *DXS* for KDE 4 and *HttpDownloadManager* for Qt 4 with certain alteration to base code for handling the layer data.
- E) Modification to *Marble* client code to correctly draw the layer data supplied to it in our custom data file format. The visual representation of data currently requires on the fly generation of thumbnails and their subsequent rendering on the layer; it would be performed by applying correct projection to the data and implementing the drawing code as QtPlugin for *LayerManagement class*.
- F) Integration with *Digikam*, *Digikam* is a prominent KDE image organiser and its integration with *marble* would make the process of geo-tagging much easier. The process would benefit from the fact that *Marble* has been coded to work as independent QtWidget too.(As described in *Digikam's* website ,there is some ongoing work to fully integrate *Marble* in it, If time Permits objective would be to complement the already work done by programmers at that moment)

Suitable GUI additions would be made, in accordance with need of user interaction with data.

Project Road map:

The whole project could be divided into 3 time frames;
Namely:

May-June: In this phase I intend to complete implementation of layers for displaying *Panoramio* geo-tagged photographs (“Task A”) and Implementation of parser for *JSON* or *JSONP* (“Task B”), but as required by the nature of tasks, “Task B” would be completed first than “Task A”.

June-July: This phase would require implementation of layers for displaying *Wikipedia* geo-tagged articles (“Task C”).

July-August: The last phase would be to implement the code which would modify the current data downloading framework to download aforementioned layer data (“Task D”), and draw the desired layer data onto *Marble*(“Task E”).

The Optional Tasks would be implemented in accordance to the time left after completing all the mandatory tasks.

Biography:

I am 22 years old, Computer Science & Engineering student from India. I was introduced to Linux & open source, 3 years ago by one of our prestigious senior professors. Since then I have been hooked to open source community in general; my contribution to community started with little steps like Linux installations and went on to lay path for major steps like founding Linux User Group in my university 2 years back.

The search for a good cross platform programming framework for my final year project (Altruism Xenophobia & Evolution of Cooperation in social Agents) led me to Qt (4.2). Qt complemented my already existing skill set and that led me to use Qt in virtually every application / program I have written since.

Moreover knowledge of Qt helped me contribute to KDE community; starting with Junior Jobs for “Amarok” and “Kturbeling”; but Google Summer of code brought an exciting surprise for me in form of “Marble”, what got me intrigued was Marble was a pure Qt4 application as well as a KDE too; combined with my passion for Computer Graphics, my desire to participate & contribute to Marble was doubled. Though my inexperience in visualization of Geo spatial data may be interpreted as possible hurdle but it’s my belief that my capability to adapt and learn would surpass any challenges / hurdles posed in course of completing this project.

My CV can be accessed from [Http://techfreaks4u.com/ShashankSingh/CV.pdf](http://techfreaks4u.com/ShashankSingh/CV.pdf)

The PDF Version of this application can be accessed from