

Bluetooth Devices Visualization

Tiago Camacho

University of Madeira - UMa
Colégio dos Jesuitas, 9000-081 Funchal
tcamacho@sapo.pt

Emanuel Fernandes

University of Madeira - UMa
Colégio dos Jesuitas, 9000-081 Funchal
emanuel.m.fernandes@gmail.com

ABSTRACT

The rapid increase of information has led to an array of problems that sometimes are difficult to handle. In the world of information that we live in today, it's become essential to find ways to filter information and present it in a meaningful and clear way. In this paper we present a way to collect data from Bluetooth devices and represent that data using a pie based chart. We will differentiate between devices that are identified and those who aren't. We will also try to define various time frames in which to present the information, so the user can have different perspectives of the information that he is viewing.

Author Keywords

Information, Filter, Bluetooth, Chart

INTRODUCTION

Various ways exist in order to represent information to a user. Some of those ways concentrate on evaluating data in real time, filter it and represent to the user. This is done so the user can have some kind of notion of what is happening in the current time. Other kinds of visualizations try to evaluate existing files of data (potentially large), manipulating them and trying to extract meaningful information. The current work is based on the usage of the second type of visualization. With this work we intend to present an historical view of Bluetooth devices that have been discovered in a certain location. We do this by reading an existing CSV type file. This file will be created by an existing application named CityWare that will be then processed by our application. The application will execute a series of filtering mechanisms in order to extract meaningful information. After that extraction the information will be presented to the user with the help of a pie based chart.

VISUALIZATION DESCRIPTION

The current type of visualization works by reading information from a CSV type file. This file, which will be written by the Cityware application, contains a set of information related to the discovery of Bluetooth devices on a certain location. The file is written in a CSV table format and needs to be translated to a graph form in order to be usable by our application. In order to achieve that we use a certain class that manages to parse the file and create a graph by selecting certain fields and computing useful

information. After that computation has been performed, the application will create a visualization that consists on a pie based chart. This pie based chart has a set of time values that be chosen. When chosen, the visualization will present the amount (in percentage) of the named and unnamed Bluetooth devices. When choosing the slice that corresponds to the named devices, we are presented with another set of slices that contain the concrete names of the devices, and where the height of those slices is propositional to the number of occurrences of a certain device. We have based our work in two specific existing kinds of visualizations, namely CoxComb [1] and DocuBurst [2].

ARCHITECTURE

We used a set of different classes in order to achieve the desired functionality. First off, the class responsible for the concrete creation of the visualization is "PiemenuVisualization". This class will hold a series of methods that are responsible for the graphical representation of the visualization. This class will make usage of various pre-existing classes that we adopted from existing work [2]. "PiemenuVisualization" will also make usage of "TableParser" class. This class will be responsible for reading the information that CityWare created. After it reads that information it will also be responsible for transforming the information by calling "TimeManager" class. This last class will have the job to hold all the time related data that was contained in the data file. "TimeManager" will be composed by a set of "TimeUnit" objects. These objects contain all the information about Bluetooth devices in a certain time unit. For example for the month "March", we will have a "TimeUnit" object that will represent all the Bluetooth information about that month.

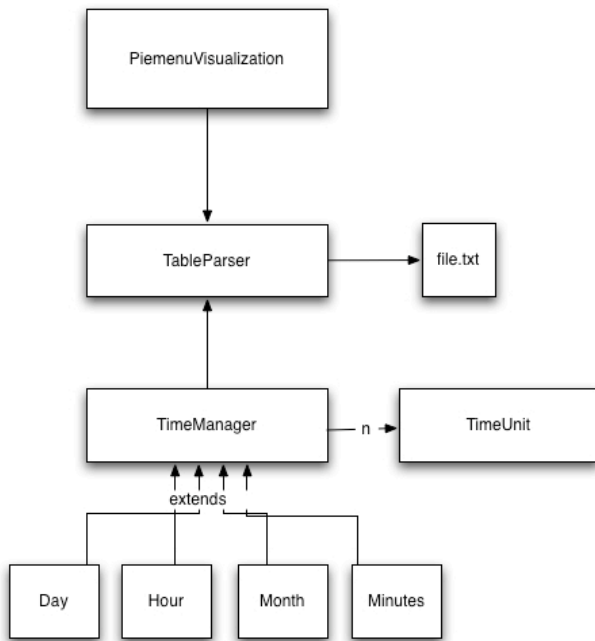


Figure 1 - Application architecture

EXISTING CODE AND MODIFICATION

The current application has been heavily based on the existing code of Collins [2]. A set of existing packages and their respective classes have been used in order to build our application. Nevertheless, code has also been added and changes to existing code have also been made. Next we present the list of files that have been added or modified:

- Modified the existing “StarburstDemo” file renaming it to “PiemenuVisualization”. Essentially we added some methods, modified lightly existing ones and added some comments in order to facilitate the comprehension of the code.
- Modified lightly the “StarburstLayout” file in order to support pie slices with different heights.

- Added “TableParser”, “TimeManager” and all of its subclasses (“Hour”, “Day”, “Month”, “Minutes”) and “TimeUnit”.

LIMITATIONS

Although we wanted to achieve a fully functional visualization, such goal wasn’t completely achieved. So, the application functions with the following limitations:

- The file that is read by the application must contain header information in order to the visualization can work properly.
- The time drill down functionality doesn’t work. Only the default one, which shows the month information, works.
- The width of each slice in the pie chart is not truly proportional to the number of existing devices for the correspondent time unit.
- The application does not handle well with big files. It is recommended to use one file with info for different months, with about 100/200 lines.

CONCLUSION

Although the visualization world is a compelling and interesting one, the leap between the idea and it’s is a fairly big. The used library for this work (Prefuse library) is a powerful but also difficult to use. Our experience with Prefuse made us realize that the best approach to use it, is to start coding up on an existing example, rather than coding from scratch. This situation derives mainly of the high learning curve of Prefuse, but also from the poor documentation. The difficulties that the group felt during the construction of the application were high, but on the other side the learning experience has been of great benefit.

REFERENCES

1. Florence Nightingale - CoxComb
<http://www.math.yorku.ca/SCS/Gallery/noframes.html>
2. Chrispother Collins – DocuBurst
<http://www.cs.utoronto.ca/~ccollins/research/docuburst/index.html>

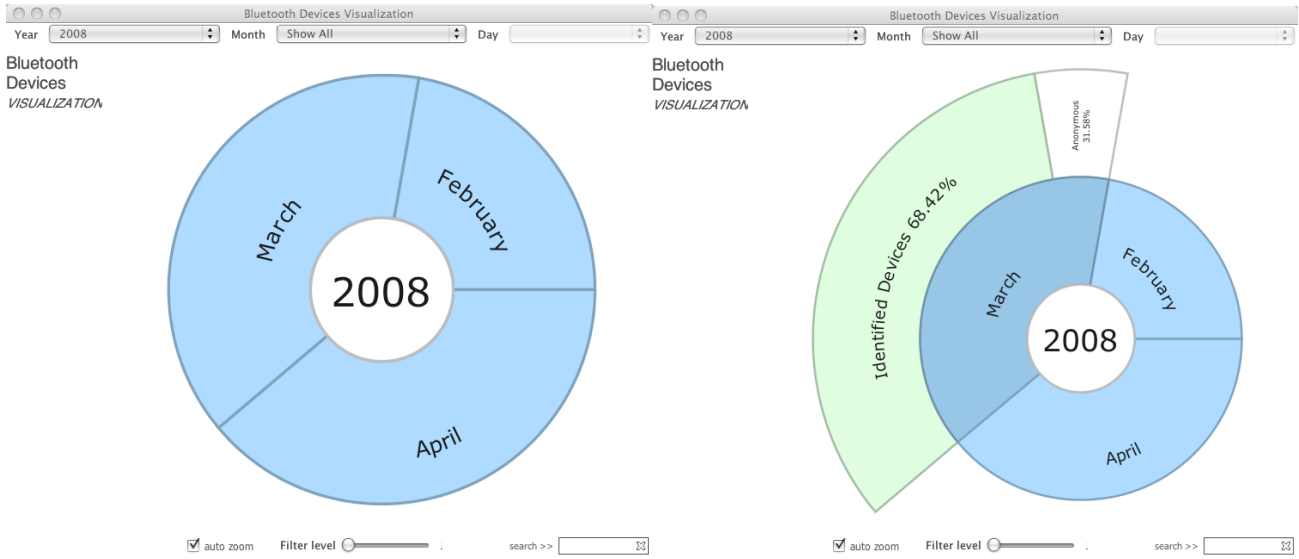


Figure 2, 3 - Piemenu Bluetooth visualization image 1, 2

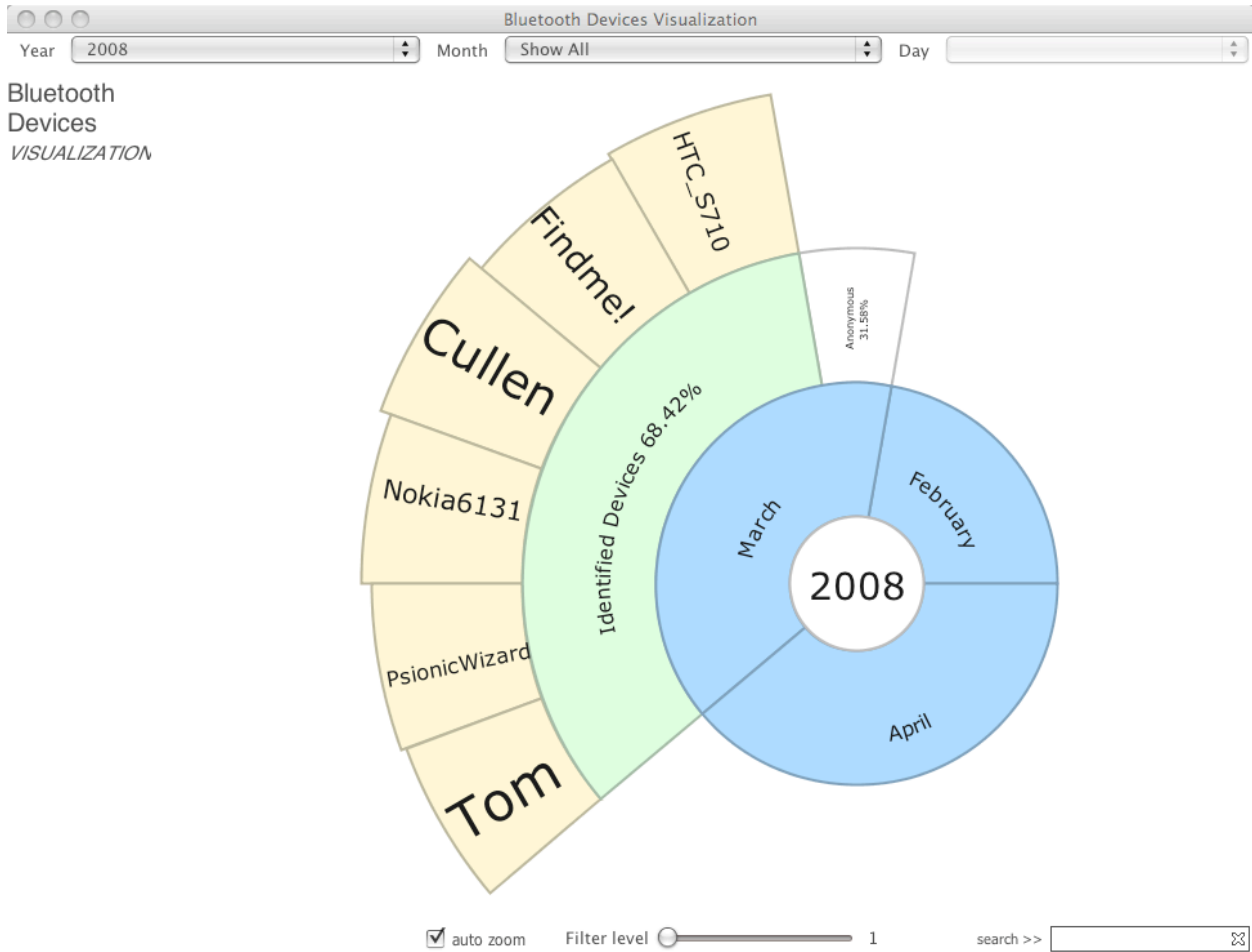


Figure 4 - Piemenu Bluetooth visualization image 4