DemoGraphr

Ambient information visualization of Flickr geo-tagged data

Florian L.P. Guitton
M.Sc. Advanced Computer Science
University of Kent
August 2012
± 11000 words

Abstract

This document and the related application investigate the concept of ambient information visualization and its application through the realization of a software program. The entire work takes its roots into the research field of graph aesthetic, information visualisation and informative art.

We will introduces the methodology used to design the visualization and drive the project development. At first focusing on data representation and then progressively switch to the artistic side of the concept.

We will elucidate the main idea of this project, which consists of transposing the idea of a time scale in family tree-like trees, from the edges to the nodes. It is about inversing the traditional shape convention of visualization based on graphs.

Keywords: AMBIENT INFORMATION VISUALIZATION, GRAPH AESTHETIC, INFORMATIVE ART

Acknowledgements

I would like to thank my supervisor, Dr Peter Rodgers, for his guidance during all the steps of this project. Giving me direction and advice when I needed it, making me reflect upon my work and choices, without which, this work would not have been possible.

I wish to express my sincere thanks to my relatives and friends as well, who encouraged me and offered their support at all times.

Table of Contents

1. INTRODUCTION

We are living in a world where data is created every single second, from everywhere and about everything. This data exists in different formats and are all relevant to a particular topic, or in a specific context.

From text, to images, to sounds; their diversity and complexity truly reflect our modern environment, and it is not anymore a problem of storing them, but a problem processing them that arose in the past few years.

There are actually several problems that arise from the over-presence of data. Indeed, more than making the most of this data we still need to understand the outgoing results. It is the role of data-mining, for more than three hundred years now (Chakrabarti, et al., 2006), to help us coping with the processing part, making the most of the data. But the situation still need our attention and our abilities to create convincing representations to properly induce a proper understanding.

1.1. THE PROBLEM

This document aims to address the problem of representation, regarding a certain context and a precise kind of data. We want to enhance how quickly and easily one can access the information hidden behind the result.

The modern world is connected. People communicate with each other using gigantic networks where everything is related to something else. It is possible to characterize these network structures as data, where each different item is linked to one or many others. We virtually represent this type of structured data as graphs.

Graphs are a well-known type of data structure, but still, most of the time visually represented the same way. Therefore, the idea is to focus on graphs, trying to sketch some new viewpoints to represent them, putting them into the context of everybody’s environment.

In nowadays life style, people want everything to be at a touch of a finger, all integrated, customizable and suited to their personality. This means not clashing with their environment. With all considerations that ambient information visualization concepts bring us, we want to provide a product with a deep understanding of these “people-matching” criteria.

1.2. The challenge

The true challenge in this project is to investigate the concept of "ambient information visualization", and pieces of knowledge that follow from it. We have to consider the subjectivity of what a representation is, according to everyone in general.

The literature will help us to frame the key points, which will allow us to detach the work from the person, and focus it onto the people. It will also help us to project an idea of "intelligence" into the communication process that will occur from the machine to the user.

We should produce an "intelligent communication" (Hayes, et al., 2008), broadly speaking with the same concerns that arise when designing interfaces and visuals for precise population (e.g. children or sense-impaired people).

1.3. Contributions

The main purpose of this work is very much to consider the link that exists between the data, the connate representation and the human beings. We will look into possible solutions that give rise to the "intelligent" aspect of the communication between the machine and the user in a context of social data as graph.

We will propose a different way to consider nodes in a graph, transposing the notion of time, from the edges to the nodes in the context of tree-like representation.

We will finally provide a fully functional implementation of the work, which will allow us to evaluate the pertinence of the work, and to judge if the exploited concepts were integrated in the best possible way.

1.4. Document structure

In the first part, we will introduce the background concepts (2 below). Then, we will approach the methodology and the development stage of the project (3 below), incrementally reviewing the evolution of the different prototype (4 below). This will lead us to the assessment of the produced work (6 below). And we will finally conclude and formulate a glimpse ahead for possible future work (7 below).

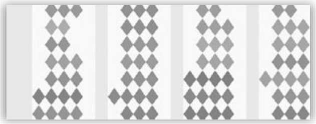
2. Background

In this section we will have a brief overview of the main concepts that drove this project which are necessary to get the essence of the thinking process. The former subpart will quickly present the concept of information visualization that we will use. Then, in a second subpart, we will reveal the concepts of graphs and will give an example of representation style that beautifully occurs in the real world. Finally we will approach the notion of aesthetic validation.

2.1. Visualization

Information visualization refers to the study of representation for large scale data, and non-numerical information (Munzner, 2008). It is defined as both an art and a science of representing information (Robertson, et al., 2009). Objectively, displaying information to the people “dates back to the dawn of time”; from the necessity of communicating ideas about abstract concepts, facts and quantifiable data. From prehistoric cave paintings (do\work Publishing, 2011) to the map of Napoleon's disastrous Russian campaign, designed by Charles Joseph Minard in the early nineteenth century (Figure 1). But the real overgrowth started with the arrival of computer graphics and the necessity to process graphical data for science in the late eighties.

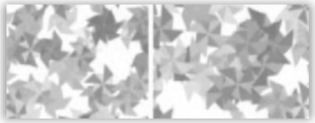
Informative art & Ambient information visualization



Some interesting representations, such as the Mondrian visualization (Figure 2), used to represent bus traffic and weather forecasts, the Activity Wallpaper (Figure 3) or as the Soup Clock (Figure 4), inspired by Andy Warhol’s work, and supposedly acting as an egg-timer, are notable examples of what informative art can be. Linking a strong sense of aesthetic and “artistic-ness” with useful information targeted to an informed population.

Ambient information visualization concept is less easy to define, as it could be understood in two different and distinct ways. Indeed, the idea of ambient information visualization can refer first to the creation of representations and visualizations, based on ambient information and data coming from sensors networks in a specific place or area. It is precisely the vision adopted in the aforementioned work on Activity Wallpaper. The idea was to compose a real-time visualization of the occupation of a given place, basing the calculation on the collected ambient noise (Skog, 2004). But on the other hand, it can also refer to the production of whatever type of information visualization outcomes displayed into the surrounding environment. It is a different kind of contextualization of the information that, again, the aforementioned work on the Soup Clock uses in an artistic way. It is a kitchen-themed visualization that represents an egg-time which puts the place and the purpose in perfect harmony. These two different visions are exquisitely completing each other and often consider as a whole.

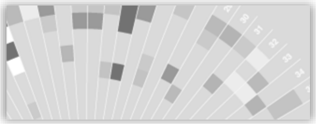
This simultaneous approach of both informative art, and visions of ambient information visualization, are increasingly investigated in a context of current important concerns. In the past few years, because of the growing awareness of people regarding to sustainable development and ecology, whether the population considers it as a fashion or as a crucial turning point in human life style, new businesses appeared and developed a strong interest towards mechanisms and tools that provide people with a real control over their household consumption (Sohn, et al., 2009). The main problem is that humans and technology are usually not interacting and communicating in a satisfying way. It is even truer with the elder generation (The Barna Group Limited, 2009). While different works around the area of digital ethnography try to understand the essence of such a phenomenon (Masten & Plowman, 2003), ambient information visualization techniques can already attach people's attention to the technology that surround them. In the precise field of domestic consumption, in terms of energy management, some attempts have been made to make energy systems give interesting integrated feedbacks to the user (Rodgers & Lyn, 2011). For instance, the idea consisted in using very simple shapes like phyllotaxis patterns (Figure 5) or pinwheels (Figure 6), arranging them in space using different layouts and colours.

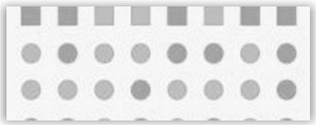


Human Perception, Culture & Issues

By its very nature information visualization relies on key human factors relative to pattern recognition, colours and motion perception, spatial reasoning or sense-making skills (Robertson, et al., 2009). Because of this strong bond between the human and a representation, a number of parameters have to be considered, if we want to produce something useful as well as usable.

When designing a visualization, we should consider the "technical" details and focus on important perceptual principles (Ware, 2004). As a reasonable example, humans are generally more precise in dealing with black and white productions when compared to coloured ones, especially when considering that people could be colour-blind, so it is crucial to make sure that visualizations are still useful and legible in black and white versions. Moreover the equipment used by the users could potentially be a problem, as it could be calibrated in a different way from one unit to another (Mc Carthy, 2002). But, even deeper into human perception, interpretations are also strongly influenced by the cultural background of the being. Thus, it is essential to pay attention to the colours used in visualization. For example, these colours could be considered in a different way from one person to another (Healey, 1996) (De Bortoli & Maroto, 2001) (Figure 7). For instance, "death" is symbolized using black in western countries, but white in China, blue for Muslims and purple in Brazil.



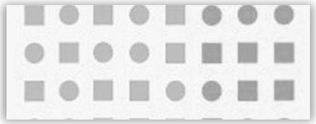


A number of key principles arose among these concerns in the past decades, such as: pre-attentive visual processing (Figure 8), and proximity or spatial stability. From a cross-field perspective, even psychologists have been studying cognitive these phenomena to create theories about perception. It is the case of the very well known "Gestalt Principles". These theories try to describe how people tend to organize visual elements into groups and define five main axes: similarity, symmetry, continuation, closure, proximity and figures (Tuck, 2010).

Pre-attentive overview

The human visual system is able to handle a large bench of scenarios when it comes to vision processing, such as recognition and detection of: colours, motion or patterns.

It is proved that some persons are able to process input signals coming from their eyes without being able to properly see, due to striate cortex lesions for example (Celesia, 2010). If such things happen, it means that a lot of unconscious processes go on in regards to perception.



The entire idea about the concept of pre-attentive visual processing, is that the brain will make sense of what it sees before the subject is even aware of what he is looking at, and starts thinking about it. We face a situation of nearly automatic response (Healey & Enns, 2012). As shown in the appendix (Figure 8), the brain is able to pre-attentively group elements more rapidly when they have the same colour rather than the same shape. Empirical studies have shown that some types of features, like colours for instance, are more likely to be pre-attentively processed, and so, are valuable when designing visualizations that should have immediate impact.

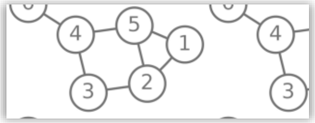
A glimpse at Gestalt principles

The human brain is indeed tempted to execute some basic actions such as grouping elements. These phenomena have been studied from the early twenties. Gestalt's laws of perception are some of the most exploited theories when it comes to structure visuals in a way that human cognitive abilities can make sense of it, extrapolating the meaning behind the picture.

These rules describe how the user tends to group the elements on a visual. Notably based on the notion of similarity, proximity, continuity and others. Proximity means that the people tend to assume that elements sufficiently close from one another are most likely to be part of the same object and so the brain interpreted it as a whole. Even more, a notion of connectedness can be added which imply that elements are being linked together, like circles very distant from one another, between which a line has been drawn, tend to be considered as a whole as well. This links to the notion of continuity which supposes that some elements are analysed the same way by the brain, because it will naturally follow a curve for instance. This will lead the eyes to a particular point in space and may induce interpretation of crossed elements. Yet again, similarity between elements on a scene will lead the brain to consider it as a pattern, so as a unique stable entity.

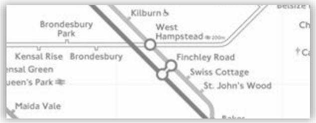
2.2. Graphs & art

A graph is an abstract construction containing a set of objects, also known as vertices or nodes, where pairs can be linked together through the use of edges. Graphs can be directed, which imply causality effect or simply flat, just linking related entities.

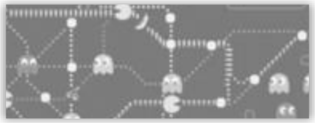


Typically, in the history of graph representation, graphs are represented as node-link diagrams where vertices are symbolized by dots or boxes and edges by curves or lines in the Euclidean plane (Figure 9). But through years and the evolution of needs, different categories of graph layouts have shown up.

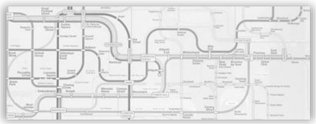
The London underground map



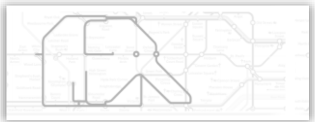
One of the best examples of linking graph representation or layout and information is the famous London Underground map (Figure 10), from which the model has been copied over and over again in dozens of countries and cities during the past decades. Designed in 1933 by Harry Beck, following the rules of his simplified topological design, this visualization is the perfect example to illustrate how information can be easily communicate to the people.



Apart from the classical appearance that we all know, a more artistic approach of the information is possible. As demonstrated by some companies in a typographic work around the tube maps (Fadeout Design Limited, 2011)(Figure 11) or even a Pac-Man themed map (Figure 12) creativity is essential in this process. But sometimes, changing the layout, considering zones as main drawing constraint, transforming the final result appearance (Rendgen & Wiedemann, 2012)(Figure 13) is another option.



Some other people worked on getting the attention of the young population towards information. With the *Animals on the Underground*, a big art project has been done to make the approach of information entertaining for children (Figure 14).



2.3. Aesthetic & Validation

When we talk about aesthetics we face a problem of strong subjectivity. To face that problem, and still be able to assess work in respect to aesthetic judgment, we have to make up some artificial aesthetic criteria based on measurable, quantifiable elements.

We can find in the literature, that very few studies have been made regarding to this problem (Purchase, et al., 1996). We also notice that, when dealing with highly subjective matter, empirical checking is necessary to confirm or infirm previous hypothesis.

It turns out, from experimentations, that two of the most crucial points in the visualization understanding of graph background data, were the quantity of edge overlap and edge bending. Despite this fact, the influence of other parameters such as symmetric-ness tends to be considered as valid as well. There is a whole range of possible new studies that could underline new concerns of this kind.

As a validation criterion, we could also be mentioned the concepts around colour selection in representation. Including cultural and medical aspects, human eventual diseases (Healey, 1996) (De Bortoli & Maroto, 2001), as mentioned for exemple.

3. Design and Method

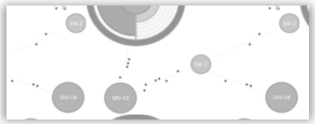
It is important to consider that this project has not been a perfect continuous process. Before the implementation work that we will detail in 4 below, ideas have been projected in order to gradually get the essence of what the final result should turn towards. In this section of the document we will demonstrate how we have been approaching the specifications of the product, from the beginning to the dawning of the first functioning prototype.

3.1. Project birth

First considerations

Before the project even starts, some ideas have already been projected as a base to surround and define the coming work.

To come up with these ideas, different practical scenarios have been pictured. The primary one was focused on computer network monitoring. For this particular problem, we can actually monitor different things: the topology, the diversity of nodes and the state or utilization ratio of these nodes, or even their characteristics. This scenario is great to imagine how we can combine all this information into a unique comprehensive and coherent visualization.



At this time, no real questions have been asked, regarding to the aspect of the outgoing representation and a quite conventional raw idea was presented (Figure 15). The principle was actually a classic graph representation, but organized in a way that allows the presentation of more information, to focus differently on the data.

As a core, we have nodes linked with lines; but nodes are mini-visualizations by themselves, displaying usage ratios of the equipment, and lines are livened up with moving animated entities. These small entities are visually migrating from one node to another, symbolizing packets of data transferred through the networks.

This idea was finally really unadventurous but has been a good start to approach the whole panel of possible story boards that come with graph data.

A more plausible scenario

Following on from the first scenario, came another one, based this time on social data. These days, social media is really popular and even if it is not that evident, they contain large amounts of information that can be interesting to monitor and analyse. However, the diversity of the data in this context is huge and it complicates the work massively.

First came the stage of social media selection. It is a quite important process in matters of interestingness and intricacy of data gathering. The approach was to leave aside, traditional text-based social media, like Twitter or Facebook (although providing picture exchange features), and focus on more visual contents, like photographs, available on services like Flickr for instance. The motivation for this choice was the eventuality of using photographic elements to build the final representation.

This finally became the starting point of the thinking process. Several things needed to be done: selecting the relevant features from the data and creating a coherent model to visualize it, something that can bring focused information to concerned people; designing a representation for this data, which is a change from the previous representation; and finally implement a first application to build these visualizations.

EARLY PROJECTION

Having outlined the scenario, it was easier to frame an idea of representation. Data available in social networks comes with time related information, and the first big step was to consider this dimension of time, and not represent the graph as objects bonded, but objects differently bonded through the time. The legitimate question at this time was: How can we only represent the connections at time τ, but retain the previous connection in the same graphical space, to see everything at a glance?

The answer came as we can represent the nodes, not as dots or bubbles, but as lines. Lines from point A to point B represent a time lapse. As can be structure a music score (Figure 16), every line on the stave is representing a node, and notes sometime link stave's lines together (Figure 17). The interesting concept in this picture, is the eventual ability to produce not only visual information, but also sonorous information, playing the produced "visualization".

The next draft was actually relating this idea. The time is going from bottom to top and the connections are represented using horizontal lines. At the bottom are highlighted the hot points that the user specifically wants to observe, the rest of the visualization is built from the data related to these hot points.

After this projection, the idea of "How to represent?" became clearer but we cannot really consider an accurate representation if we do not consider "What is represented?" and "Who is this targeting?" It is precisely the next questions that arose.

DEMOGRAPHR

We gave a ruling on social data, coming from the popular Flickr, and related it to time. But pictures from Flickr come, for more and more of them, with geographical information as well. And it is from this fact the final use case came out.

The application is about to create visualization for "demographers" and people wanting to study or just visualize the movement of the modern connected population. Through the thousands of pictures uploaded on Flickr every hour, a visualization is built, symbolizing travels of people between countries. Consequently we decided to inspire the name of this application from the notion of demography, the link with data as a graph and the Flickr name giving: *DemoGraphr*.

With this idea, the design still need a bit of "artistic-ness" to fulfil the requirements of ambient information visualization. The previous draft was promising and a second one finally came. We decided to change the model for the overall shape of the representation to adopt a tree/mushroom shape (Figure 18). This interpretation as a natural element will give the representation a kind of abstract artistic aspect, which fits the concept of ambient, environmental décor. We will then, along the implementation process, level different aesthetic criteria to transform this idea as necessary.

3.2. Requirements and Specifications

From the beginning we wanted the final application to be easy to use and accessible. We decided to base the work on a multi-platform solution, providing the user with different usage options.

ActionScript 3

Despite the fact that ActionScript 3 is a language fully supported by Adobe Systems Incorporated, it is actually an open-source language with accessible specifications. Being a dialect of ECMAScript (Hoehrmann, 2006), ActionScript 3 is a very legible object-oriented language designed for controlling 2D and 3D vector animations. This precise characteristic makes ActionScript 3, in a context of visualization production and graphic drawing, a perfect candidate for this project.

Moreover, the language is supported by the Apache Software Foundation, which provides a free open-source complier, Apache Flex, formerly known as Adobe Flex (Apache Software Foundation, s.d.); as well as Mozilla which develops an open-source virtual machine called Tamarin (Mozilla Developer Network, 2011). This openness allows us to realize the application in a perspective of multi-platform usage which validate our requirements.

Visualization structure

Following the projection made earlier, we tried to define the different components of the representation and how they should interact visually with each other. As the visualization should adopt a tree-like overall structure, we will consider the core structure as an association of lines, vertically parallel at their bases and slightly bending when moving to the top, showing and "exploding effect". Each one of these core lines will represent a node of our graph, for instance a country, and will be represented in a black shade.

On top of this core ensemble, which will result in a symmetric shape, we will draw some coloured arcs of a circle. These will only be present on the upper part of the core shape, above the bending point, "connecting" the different elements of this core structure. These curves are meant to represent the edge of our graph, for instance they will symbolize a journey, the trip of a particular user from a country of departure A, at one extremity of the curve, to a country of arrival B, connected at the other extremity of the curve.

Because the visualization is meant to be neat and clean, we do not attempt to display the orientation of the travel. This will be achieved by adding some interaction onto the representation. A touch or a cursor hovering over a specific curve will show the detailed information in a separated text field, with the departure and arrival point and the date. To feel this notion of time when representing the data, we will draw the curve onto the core structure considering that the bending point is a start moment in the time and while going along the lines towards their upper extremity, the time go forward. This will be done using a regular scale, so a day will be represented with the same length all along the lines for each line. It means that we will base the visualization onto arcs of concentric circles, where the bending point of the core shape will be the centre of the circles.

We consider this projection as a reasonable start to undertake the process of producing a relevant visualization, since we already consider important aesthetics aspects as non-crossing of edges and symmetry (Purchase, et al., 1996).

Congruent data structure

According to our needs, we decided to make the data structure that we will use, simple and clear. We based our work on an object-oriented data structure. Records will be grouped by user, so each user will be represented as an array of entity. These entities contain all the selected picture related information: picture identifiers, geo-tag markers and shooting date. These entities can potentially contain other information for development in future work.

3.3. Development approach

Software engineering constitutes a set of discipline which are systematically involved in the production of software content. It can be divided in fields like: software design, software testing, software maintenance and other (Abran & Moore , 2004). These disciplines generally constitute steps which are applied in a systematic and disciplined way. They represent quantifiable design approach, development operations and studies approaches.

One of the key element is software development, which is a subset of software engineering disciplines. We decided to focus on that specific part since we do not have needs in regard to a marketing or a deployment perspective. This subset groups all the basic required element that allow us to drive a development project, form design, to code, to test and validation.

It exists numbers of way to drive a project and we decided to investigate the different existing, well-known methodologies to make the most of our given time. The creation of methodologies for development driving started right in the 70s with a so called “structured programming” method, massively focus on the use of code abilities and subroutines such as loop and block structures. It quickly evolved and we saw more and more methods coming year after year. In the 90s, object oriented programming, being investigate since the 60s, finally become a major development method, which opened the door to a new variety of more human-friendly software vision.

During this evolution process, born very well-known techniques such as Scrum, qualified as an “agile” method, based on massively iterative and incremental development, where the requirement and solutions evolve through collaboration of different self-organized team. It is a good approach, but it did not suit our needs knowing that it is focused on adaptive planning, and we have a very well define time schedule. However, we have been strongly interested in the set of methods, qualified as “extreme programming”, even if a whole bunch of criticism arise from it, such as a lack of structure in the code or documentation production “in-development”

We decided to consider a “Test-Drive Development” method, in the family of “extreme programming” methods, to create the application. It is a software development process that significantly improves the product quality replacing the traditional “Code-Test-Debug” approach by a “Test-Develop-Modify” one, even if the product structure may suffer from untidiness.

This process has the property to take in consideration the chaotic aspect of specification, changing over the development timeframe, allowing a real time control over the non-regression of already developed functional aspect. Thus it eliminates numerous bottlenecks related to the development process, and contributes to stabilize the complexity of the product. As a consequence, software quality, sustainability and scalability are the catchwords of this process that allowed safer and more efficient production.

As the project ends, we will consider a refactoring of the code and document it, allowing better understanding of sources in the future, filling in the lacks induced by this development approach

4. Implementation

In this section of the document we will review the implementation process which is a very crucial part of the entire project. Despite the designing process being valuable, implementation will result in real and functioning work.

It is important, in a context of test-driven development, to get a full sense of how important the implementation step is. It is not only the process of producing code, but also a process of constant review to change the orientation of the project and assess carefully the relevance of the work.

We decided to divide the work into three parts. First focusing on data gathering, which will allow us to easily create the visualization in the future, not worrying about the data being sufficient or not. Then, a quick process of data preparation, to structure the data according to a defined format. Finally, and most substantially, we will work on the production of the representation itself, carefully assessing step-by-step the product to make directional change.

4.1. Data gathering & preparation

As a first step in the development we had to carefully think about the best way to put it in place. We started by choosing the communication protocol we wanted to use, followed by the selection of the desired methods, and finally carried out the code.

Choose a communication vector

The first step was really to tackle the documentation of Flickr's Application Programming Interface (Yahoo! UK Limited, s.d.), to make sense of the methods provided and the way to access it. Flickr provides different communication vectors that can be used to access the data: extensible mark-up language remote procedure calls (XML-RPC), simple object access protocol (SOAP) and representational state transfer request (REST). In the context of our application we used the REST API. This is considered as the best solution since REST APIs, also known as "RESTful" web services, are URI-driven hence easy to manipulate using simple requests.

Register a new application

From the documentation we also found that the use of Flickr services depend on a notion of API key, which is a unique identifier used by the services to recognize a certain application, and applies some internal policies. It allows Flickr to control the amount of data the application can retrieve in a given period of time, as well as some authentication process to access private information.

As far as this project is concerned we do not need to access private user details or picture streams, but we still need to register the application and get an API key. We finally declared the application as DemoGraphr and got the following identification key: `0e90154ec87636986750e2a584e94a1c`.

Select the Flickr's API methods

The work was then about attentively selecting the methods that best suit our needs. We wanted to have precise control over the records we retrieve, so we decided to simply use the `flickr.photos.search` method. We considered the different parameters and finally set them as follows:

```
"method"            = "flickr.photos.search",
"privacy_filter"    = 1,
"accuracy"          = 3,
"safe_search"       = 1,
"content_type"      = 1,
"has_geo"           = 1,
"per_page"          = 100,
"page"              = [the_required_page],
"min_taken_date"    = [the_required_early_date],
"max_taken_date"    = [the_required_late_date],
"extras"            = "geo,date_taken"
```

In this given set of parameters, `privacy_filter` is defined to retrieve public pictures only, `accuracy` to get the picture with a location known with a minimum precision of the country, `safe_search` to lock on safe pictures avoiding sensitive content in case of them being used in the future, `content_type` to retrieve only photographs, `has_geo` to get only geo-tagged data, `per_page` to limit the amount of data retrieved per request, `page` to ask for a particular page in the result set, `min_taken_date` and `max_taken_date` to set the boundaries in the time and finally `extras` to add in the response the geographic and shooting information.

Besides, we figured out that Flickr was sending as geographic information some identifiers that require to request the service again to get detailed information. Indeed, the geographic information is available in two alpha-numeric formatted identifiers and it is necessary to use the `flickr.places.getInfo` method to know in plain text what the corresponding country is. These requests are much simpler than for search, since the only parameter required is the previously retrieved identifiers. As follow:

```
"method"            = "flickr.places.getInfo",
"woe_id"            = [the_retreived_id]
```

Also, we selected the type of return format that we wanted among Flickr's options. By default, Flickr returns XML data, but since we are using an ECMAScript based technology we chose to request for JSON data. JSON stands for JavaScript Object Notation and it just so happens that Javascript is itself another subtype of ECMAScript, thus the objects are structured the same way and so AS3 can immediately interpret the responses. We achieve that by simply adding extra parameters to our query structures. In the meantime we added our application key as well, as follows:

```
"api_key"           = "0e90154ec87636986750e2a584e94a1c",
"format"            = "json",
"nojsoncallback"    = 1
```

Where `format` defines the type of return and `nojsoncallback` state that we do not want a JSONP-like answer, which contains a framing call-back function and it is not part of the object.

Retrieving workflow

The retrieving workflow is simple and based on a couple of actions. At first we consider the time interval given by the user, and we loop upon it to cut the time space in slice of one day. This helps us to prevent side effects of Flickr's interfaces that allow to retrieve a maximum of 4000 records per search. We empirically discovered the limitation as we will discuss in section 5. Indeed, cutting the time space will result in less records per search, so we maximize the relevance of the overall amount of data. We run all these queries in parallel to minimize the amount of time taken to retrieve the overall result. For instance an interval of 10 days will produce a chain of 10 simultaneous search queries. To do that we substitute the `min_taken_date` and `max_taken_date` parameters of our search frame, as seen in the previous part, by the proper dates calculated while looping.

Then we consider each of these queries separately. A system of counters allows us to keep a track of each query's advancements so we are able to know when the process is finished and go on to the next steps. For a particular query, once the response is given by Flickr, we store the objects in memory and read from the result, the current page number, and the total number of pages in the result set. If we are not at the end, we start a new query asking for the next page of results. However, since the maximum retrievable number of records is of 4000, and that we set the number of results per page to 100 we know that if the total number of page answered is more than 40 it means that the upcoming query is pointless, so we do not process above that. Indeed, even if Flickr does not allow us to retrieve more than 4000 records, it still shows us the real number of results in the database but keeps on giving us the same set forever from the 40^{th} one.

After all queues finished and the records are all stored in the memory, we loop across all the records to collect location identifiers and ask for the corresponding countries. We figured out that there is no unique identifier for the countries, but still, sometimes the same location identifiers are used for different records. So we decided to store the associative map of results into a separated object in the memory, which allows us to have better control over the queue of queries to avoid asking for the same location identifier twice, and to merge them afterwards during the data preparation process.

MERGE THE COLLECTED INFORMATION

The merging process is very light. It consists of taking all the raw data retrieved from Flickr servers and merging them into an exploitable format. We basically construct a third object in which we organize the data from the first one, following the structure we designed in the first place, substituting in, at the same time the location identifiers by the corresponding values in the second one.

Once this process in completed, the resulting object is seen as a provider that will feed the representation part of the application.

4.2. REPRESENTATION PRODUCTION

Probably the most interesting part of the project, as well as the most complex one. The implementation of the application part that creates the representation has been a strongly granular work. Thanks to a test-driven development method, this work remained enjoyable and organized. The overall stage produced six different versions of the application, seen as prototypes, evolving through the time. Huge paradigm changes have been made from the first design projection, to the final result. In this part we will review the reasons of each evolution and the problems we met from the realization of the projection based prototypes till the end. Later we will describe the final code structure.

ADAPTATION OF THE PICTURED DRAFT

The draft projected during the design process resulted in the first real implementation of the application. The results have been obtained quite quickly, which was a good thing, considering that the assessment of this first product was decisive to state if the chosen direction was the best one.

The outgoing representation turned out to be quite convincing (Figure 19). The overall result was neat, clean and legible. Indeed based on the common aesthetic validation criteria, as covered in section 2, the representation was valid despite a small overlapping problem of the edges when the quantity of information is important. Unfortunately, the product was not as close to our guidelines as we thought it should be. The representation had a strong lack of “artistic-ness” from a subjective perspective, so we decided to take a turn for the better, according to our objectives.

Direction toward "artistic-ness"

Although our draft implementation was convincing, since the result did not fulfil our objectives, we decided to keep the basic idea of a tree, keeping the same notions for the nodes, but redefining our vision of edges. Till that moment, the reading of our representation relied on immediate vision of connections, direct curves from one point to another. We thought it would be an interesting thing to explore the path of indirect understanding, making the user think about what he sees to get the information. In that way we can introduce much more liberty in the drawings, keeping the production legible. All the more so as it contributes to communicate one of the principal concepts of ambient information: if you do not know, you cannot get.

Consequently we projected a visualization where the previous core structure was behaving in the same way, though we introduce a notion of randomness, and where the edges would be transformed into bubbles seen as leaves. To make the correspondence between the elements, and so work out the connection, we decided to add a colour-coding system: the blue bubbles are linked with the blue branch, the green ones with the green branch and so forth. We saw randomness as a way to get more natural and organic shapes, giving a particular dimension of environment and living, dynamic, representation. We finally end with a second version of the representation.

The idea to decipher the production stays fairly simple. For country A coloured in green, and a country B coloured in pink, if we take the green branch and meet on the way, from the trunk to the extremity, a pink bubble, it means that at this moment in time, some Flickr user travelled from country A to country B. If we had a country C coloured in maroon and we met a maroon bubbles onto the green branch, it will mean that someone travelled from country A to country C.

We appreciated the production (Figure 20), which was subjectively not answering our problem, but has introduced this artistic side we were looking for. We decided to keep going on that path, but solving the problems regarding to aesthetic validation rules.

Refining step-by-step

Considering the new direction we took to represent our data, we had to review one-by-one the aesthetic criteria we considered as crucial.

At first we assessed that despite the welcomed "artistic-ness" we were looking for, we still needed to make sure that the idea of a tree stays as predominant. A very disturbing effect in the second version, in that perceptive, was that branches were sometimes growing towards the bottom of the tree, even under the trunk base. This was not seen as natural and even more, the version could produce representation with branches all growing towards the base, blowing away at the same time the overall tree shape of the visualization. We decided to allow the growing of branches with a restricted beam of π radian. In the meantime we added text, showing the country's name, using the same colour-code as the corresponding branches and leaves, on the side of the tree. This allow the user to work out the relations between the representation and the real world. Moreover we added some background branching, very faded, to accentuate the art side of the production. We came up with a third version of our application. This time perfectly legible, since we added the text, but still showing some aesthetic problems with nodes overlapping.

As our nodes are actually represented as lines, we face the well-known problem of overlapping. To avoid this issue we had to think about different solutions. Our solution generates the branches using a massively random process. We had to change that, so we could control the trajectory of the branches, therefore the overlapping. We tested three different techniques, ending up with three different 4th version of our application:

- Adjustment post-drawing
- Force based repulsion algorithms
- Incremental reduction of angles

The most immediate solution was to adjust the position of the branches, once drawn, so we did have to change the random drawing algorithm. The idea was to test, using the `hitTestObject()` method of AS3 graphic containers, all our branches objects against one another, to look for overlap and then slightly change the angle of one of them, keeping the constraint of π radian opening, repeating the process till no more overlapping is detected. It turned out that the `hitTestObject()` method is not the optimum manner and had side effects that caused the adjustment to infinitely loop resulting in trembling branches.

So we got on to a more clever solution, inspired by force base graph layout. The idea was to consider every branches extremity; including branches' top, connection points between branches and the trunk and tree's crown and base; as point on a separated sub-graph and compute a forced-based version of this sub-graph. All points are seen as handles on which the branches are attached, moving as the sub-graph adjusts itself (Figure 21). We also had to integrate a notion of fixed points, so we can force the trunk and connection point to stay still. We finally used the following very classic algorithm:

```
set up initial node velocities to (0,0)
set up initial node positions randomly

loop
total_kinetic_energy := 0
for each node
      net-force := (0, 0)
      for each other node
            net-force := net-force +
            Coulomb_repulsion( this_node, other_node )
      next node

      for each spring connected to this node
            net-force := net-force +
            Hooke_attraction( this_node, spring )
      next spring

      this_node.velocity := (this_node.velocity +
                             timestep * net-force) *
                             damping
      this_node.position := this_node.position +
                             timestep * this_node.velocity
```

```
            total_kinetic_energy := total_kinetic_energy +
                                    this_node.mass *
                                    (this_node.velocity) ^ 2
    next node
    until total_kinetic_energy is very small
```

Since this algorithm is described in numerous papers in literature, we are not going to detail it properties and functioning here. In the perspective of our project, we saw some problems appearing with the implementation of the algorithm. Our previous rules about the specific π radian beam opening got overridden by the algorithm. A solution to that problem would have been to adapt the algorithm, adding some constraint bases on the angle between the branch and the trunk. But the problem of angle extrapolation was too complex to be solved in the time we had using the tools available so we decided to try a last approach.

This third approach consisted in calculating the final inclination of the branches, before the drawing. It appeared during the two previous trials that adjusting after was not the best option, so we thought about doing it the other way. To achieve that we considered the π radian beam as two parts, one for each side of the trunk. Dividing this half beam by the number of countries to represent, getting the angle thread, randomly altering it and placing branches on the right or left of the tree with equal probability, we ended with a very simple system that reached our goal.

Thus, we reached a determining point in our implementation process. The product was finally what we expected (Figure 22). However, we still did care about the leaves aspect, introducing different size and state, with plain and empty leaves, following the very essence of nature, amplifying the idea of “realistic-ness”.

5. CORRECTING DEVELOPMENT GAPS AND POLISHING

As we used a “Test-Driven” approach, we introduced into the realisation a number of lacks, inherent to the use of this method. We focused our corrections on two different aspects: code structure and program documentation. To answer the former problem, we used a “refactoring process” and we simply commented the code, in line, to answer the second.

Then, after the code being cleaned, we decided to add our last feature, to put the user in the centre of our conception, providing easy to user configuration and demonstration abilities base on the use of an XML file. Finally we entered a bug tracking process.

5.1. REFACTORING AND USER EXPERIENCE

In our optic of “Test Driven” development, we have been changing the code of our application several times, resulting somehow, in quiet messy source code. One of the common solutions when the process is not too complex and long, is to refactor the code so we can end with a program with similar functions but better internal structures.

From this start point the application was almost relying on a single class holding all the processes of the application, from trunk growing to branch placement and connection calculation. See the UML diagram before refactoring in the appendix (Figure 23).

The goal of a refactoring process is ultimately to improving the design to make it easier and easier to work with. It enhance the code readability, maintainability and extensibility while reducing the complexity. Here again, different techniques can be used to achieve such a task, focused on three different aspect: abstracting, exploding and naming (Fowler, s.d.). Whilst we do not need of abstraction into our project, the two other aspects are quite valuable. Exploding consists in breaking the code in logical parts and naming consists in using relevant names for classes, methods …

We decided to structure it, expanding our application, one separated class for each entity. We considered as entities: <branch>, <leaf>, <trunk> and <background>. All the three handled into a <tree>. Aside from that, other classes would handle the <data> gathering, the <event> management, the text <panel>, the <initialization> of the application and some <utilities> functions that help to avoid redundancy. This brought us to 10 classes. Some of them, as the text board class, are implemented as singletons to avoid multiples declaration of the object and others are not instantiable and contain static methods only. Along that, we carefully named each classes and methods, to reflect their role and position, using the prefix “DG”, standing for *DemoGraphr*, for each class, and using a CamelCase notation for methods.

The user in control

For us is was pretty clear that producing a good application was not only done by just achieving a convincing representation of our data. The user has to be in control of it. Since, for our previous versions, we were using in-code selection of countries with fixed parameters, we decided to push the boundaries further away, introducing a configuration mechanism, based on an external file. So we built a supplementary class for <configuration> that was able to extract data from an XML file, located in the same location as the application, to override some predefined internal parameters. In that way we give total control of the application to the user.

We structured our file in a straight forward way as follows:

```
<config>
      <demo enabled="true" delay="10" />
      <option baseLength="50"
                    baseThickness="20"
                    decreaseIdxLength="0.9"
                    decreaseIdxThickness="0.7"
                    treeStartX="450"
                    treeStartY="800"
                    panelStartX="1200"
                    panelStartY="800"
                    textSpaceWidth="200"
                    textDefaultSize="10"
                    maxSimultaneousPlaceQuery="25"
                    analysisInterval="1"
      />
      <list>
             <country name="France" />
             <country name="Canada" />
             <country name="Spain" />
             <country name="United Kingdom" />
             <country name="United States" />
      </list>
</config>
```

All the information about each field could be found in a sample files that contains all possible changeable parameters. We designed our class to tolerate when parameters are not provided. For instance the user can simply skip the demo node of the file, while changing only two attributes of the option node and providing his country list. Default values are applied for everything else. However, we do not provide reconstruction techniques to deal with non-correctly formatted XML files. The use of it is simple, as we normalized all the units to use only measurement in pixels, seconds or days. The user is in control of the position of the elements as well as the selection of the data producing the outcome.

In the meantime we integrated a reloading mechanism, which check for update in the configuration file to report the modification in real time, while the application is running, onto the layout and representation. Because the visualization is going through a repeating series of actions, what we consider as real-time here is actually happening once a loop, after each rendering of the representation, or after the time you set as demonstration updating delay, as discussed below.

Getting ready for demonstration

Another important thing to think about was the necessity to provide the user with a relevant demonstration mode. The application is supposed to be entirely connected to the internet, gathering information directly from web services. But in some use cases it is likely that no internet connection will be available, for demonstration purposes for instance. Sometimes however, even though an internet link is up, the bandwidth does not allows the user to get responsive behaviour from the application, but waiting and loosing incredible amount of time, here again, for demonstration purposes.

Thus we decided to add into the application, encapsulated into a <test> class, some kind of a demonstration mode, embedding pre-recorded sets of data, playing them back continuously at a given interval, to render the representation. This allows the user to show off the production without having to wait for the data.

Bug tracking

After that previous step we ended with a sixth version of the application and decided, due to time constrains to focus on debugging and polishing the program and not adding any new features. It is indeed quite a considerable risk to add functionalities at the last moment, even in a continuous test driven approach, because we obviously may introduce weird side effects and undesired guise.

5.2. Documentation

Once the implementation done, a final step was necessary in order to deliver a comprehensive program. At a given point someone may need to used and modify the program therefore it is worth considering creating documentation or simply commenting the elements themselves. Moreover extracting a UML diagram (Figure 24).

It is the work we did for the last bit. The entire code of the program has been commented on, so any person looking at it afterwards can easily tweak the application and work out how the program works. At the same time, a sample configuration file that has been fully commented as well, allows the user to get full sense of the parameters as well.

6. Assessment and Results

Once the tough implementation work was achieved, it was time for us to evaluate the production according to some aesthetic validation criteria.

6.1. Performances

We considered measuring the performance of the representation using quantitative values. For instance we decided to run the application ten times, using three different list of countries. With 5, 10 and 15 countries; considering it respectively as small, medium and large datasets. For each run we count the number of nodes overlapping, the ratio of branches grown above one third of the trunk, the average colour distance between every branch to another and the ratio of branches growing in one side or the other of the tree.

The measurement of overlapping is very easy, and pretty straight forward, just consisting of manually counting the intersections.

In the same way, the proportion of branch growth above one third of the trunk is calculated by hand, based on the number of pixels from bottom to top, not considering trunk inclination and based on the connection points between the branch and the trunk.

The colour difference is measured using a tool called Colour Contrast Analyser (Faulkner, 2008). For this precise assessment, we are not considering diseases related to anomalous trichromacy or complete dichromic deficiency such as tritanopia, tritanomaly, protanopia, protanomaly, deuteranopia and so forth. We evaluate the colour contrast based on normal vision, following the World Wide Web Consortium contrast ratio algorithm. We consider a result above 2.0 to consider the representation as valid.

The last criterion is again of the same nature as the former, since we manually count the branches on the right and left side.

Small dataset

	Overlapping	*Grow point*	*Colour Δ*	*Repartition*
1	0	20 %	2.81	4/1
2	0	60 %	2.04	2/3
3	0	20 %	2.10	4/1
4	0	40 %	1.89	4/1
5	0	20 %	2.91	3/2
6	0	40 %	1.74	1/4
7	0	20 %	2.03	2/3
8	0	40 %	1.05	1/4
9	0	40 %	1.50	2/3
10	0	80 %	2.47	3/2

Medium-sized dataset

	Overlapping	*Grow point*	*Colour Δ*	*Repartition*
1	0	30 %	2.20	7/3
2	0	20 %	2.61	6/4
3	1	30 %	2.73	5/5
4	1	30 %	2.18	5/5
5	0	40 %	1.82	8/3
6	0	50 %	1.74	6/4
7	0	40 %	2.08	4/6
8	1	30 %	1.94	4/6
9	0	30 %	2.21	5/5
10	2	70 %	2.07	2/8

Large scale dataset

	Overlapping	*Grow point*	*Colour Δ*	*Repartition*
1	0	53 %	1.60	10/5
2	1	46 %	1.72	6/9
3	1	40 %	1.91	5/10
4	2	46 %	1.20	7/8
5	3	46 %	1.23	5/10
6	1	40 %	1.34	6/9
7	2	46 %	1.31	7/8
8	0	40 %	1.12	7/8
9	1	53 %	1.48	8/7
10	0	40 %	1.46	8/7

As demonstrated, we compacted the measurements into tables from which we made some observations.

We supposed here that every run had been executed with the same exact parameters except the country selection. It is very important to notice, since it is possible to change dimension, length and thickness of the elements on the visuals to make the representation re-enter into our validation limits.

Regarding the overlapping, the chosen technique reveals some good results, with almost no overlapping for small and medium datasets. Though it started to increase above this. Oddly, the percentage of branches growing above one third of the trunk remained around 40 % although the trend legitimately seems to increase as well. The outcome for colour validation was however a bit predictable. Since colours are generated randomly during the process, it is likely that very similar once are chosen. We clearly see the level of clarity going down although it seems that a fair amount of small and medium size sets still seems to reach our validation minimum of 2.0. Finally the repartition of branches is stable and balanced aside from a few extreme cases.

6.2. PROBLEMS

According to our evaluation of the program performance, the visualization is relatively good, but the problem of colour selection remains. We should consider in the future to use a specially designed algorithm, which could selection the colours in a clever way to maximize the visual separation of the items, allowing for more of the contrast evaluation averages to pass the threshold.

Beside the assessment problem of the project, we have been continuously facing all along some difficulties inherent to the tools we were using, and some related to concerns we did not anticipate.

From a technical perspective, and as previously discussed, Flickr services do not allow us to retrieve more than 4000 records per search. Even if you did bypass this a bit, the problem it is still an issue. Otherwise, the coding technology we are using and in particular the Flash Player runtime environment, which, for instance, implements some safety precautions to avoid infinite looping. This detection is based on the time taken by a specific method to complete. Unfortunately, processing huge amounts of data may take time and the runtime virtual machine may shutdown the application. Still, we have been pushing this limit at the compilation up to 65535 seconds, which is the maximum possible value as contained into a short primitive type.

From another perceptive, since we did not consider enough problems around vision deficiency and colour-blindness, the application turns out not to be useful addressing this portion of the population. Flatted into black and white as shown in the literature (Robertson, et al., 2009), a given visual should still produce a legible output. This is not the case due to the use of a colour-coded communication.

7. CONCLUSION

The project finally led us to an interesting output. Even if the representation is based on a tree, as countless before, it was interesting to contribute using the branches as time scale, like in family trees, but also as nodes. We did not find much of the same concept in the reviewed literature. This makes the most significant contribution of this document: Rising up this idea of transferring the time information from the edges to the nodes themselves.

Besides, it gave us a fair awareness of the human problem, highlighting certain points that must be considered every-time such as colour-related concerns arise. Of course this has been underlined quite often in the literature, but unfortunately is still not fully integrated, as humans designing the system are usually not the ones that use it, and certainly not all-knowing at a given time.

7.1. EVALUATION

From a personal point of view we considered the project as very rewarding. It is not just a question of technical background, but awareness of, biology, cultural differences, ethnography, art, technology, history or human perception. A rich multidisciplinary exploration of knowledge that opened up our general understanding of interaction and communication.

7.2. DIRECTION FOR FUTURE WORK

The obvious thing to think about when planning for further investigation, is to solve the already known problems as well as improve the mechanisms used.

Technically, different improvements could be made. For example, adding a real queue management system could be a good step forward to avoid link bottleneck and timeout. Finding a best way to deal with the Flickr search limitation could be a significant step as well, providing more reliable data to work on. Even more so considering a new coding technology could help avoid runtime machine problem such as safety timeout for long loops.

For convenience purposes as well, we can aim to provide a configuration interface that could produce or modify configuration files. Moreover enabling the program with natural interactions, to manipulate the visualization using touch.

On the other hand, it could be interesting to run a survey, with a panel of volunteers to whom questions could be asked, in order to determine the impact of the visualization upon them and define the major features that make a difference in understanding the information. A qualitative assessment could be done based on the observation of tree evolution in art history or even from a cultural aspect.

8. References

Abran, A. & Moore , J., 2004. *Guide to the Software Engineering Body of Knowledge.* s.l.:s.n.

Alt, F. et al., 2010. *Exploring Ambient Visualizations of Context Information,* s.l.: s.n.

Apache Software Foundation, n.d. *Apache Flex (Incubating).* [Online]
Available at: http://incubator.apache.org/flex/
[Accessed July 2012].

Beckman, B., 1994. Theory of Spectral Graph Layout. *Tech. Report.*

Brandes, U., Harel, D. & Koren, Y., 2009. Spectral Graph Drawing. In: *Handbook of Graph Drawing and Visualization.* s.l.:s.n.

Card, S. K., Mackinlay, J. D. & Shneiderman, B., 1999. *Readings in Information Visualization : Using vision to think.* 1st ed. San Francisco: Morgan Kaufmann Publishers.

Celesia, G., 2010. Visual perception and awareness. *Journal of Psychophysiology,* XXIV(2), pp. 62-67.

Chakrabarti, S. et al., 2006. *Intensive Working Group of ACM SIGKDD Curriculum Committee.* s.l., s.n.

De Bortoli, M. & Maroto, J., 2001. *Colours Across Cultures: Translating Colours in Interactive Marketing Communications.* s.l., s.n.

do\work Publishing, 2011. *Data Visualization - Convey, Clarify, Construct.* I ed. Hong Kong: do\work Publishing.

Doğrusöz, U., Madden, B. & Madden, P., 1997. Circular layout in the Graph Layout toolkit. *Symposium on Graph Drawing,* pp. 92-100.

Fadeout Design Limited, 2011. *Trnsprtnation art for Transit Enthousiasts.* [Online]
Available at: http://www.fadeoutdesign.com/index.html
[Accessed July 2012].

Faulkner, S., 2008. *Contrast Analyser for Windows and Mac.* [Online]
Available at: http://www.paciellogroup.com/resources/contrast-analyser.html
[Accessed August 2012].

Fowler, M., n.d. *Refactorings in Alphabetical Order.* [Online]
Available at: http://refactoring.com/catalog/index.html
[Accessed September 2012].

Grinstein, G. G. & Levkowitz, H., 1995. *Perceptual Issues in Visualization.* 1st ed. Berlin: Springer.

Hayes, G., Monibi, M., Patterson, D. & Kaufman, S., 2008. *Interactive and Intelligent Visual Communication Systems.* Chicago, Association for Computing Machinery.

Healey, C., 1996. *Choosing effective colours for data visualization.* Los Alamitos, Institute of Electrical and Electronics Engineers Computer Society.

Healey, C. & Enns, J., 2012. Attention and Visual Memory in Visualization and Computer Graphics. *Institute of Electrical and Electronics Engineers Transactions on Visualization and Computer Graphics,* VII(12), pp. 1170-1188.

Hoehrmann, B., 2006. *RFC 4329.* [Online]
Available at: http://tools.ietf.org/html/rfc4329
[Accessed July 2012].

Kempf, K., 1961. *Historic Computer Images : The Computer Tree.* [Online]
Available at: http://ftp.arl.army.mil/ftp/historic-computers/
[Accessed August 2012].

Koren, Y., 2003. On Spectral Graph Drawing.

Masten, D. & Plowman, T., 2003. Digital ethnography: The next wave in understanding the consumer experience. *Design Management Journal,* XIV(2).

Mc Candless, D., 2009. *Colours In Cultures.* [Online]
Available at: http://www.informationisbeautiful.net/visualizations/colours-in-cultures/
[Accessed August 2012].

Mc Carthy, A., 2002. *Color Imaging Workflow Primitives.* Scottsdale, International Color Consortium.

Mozilla Developer Network, 2011. *Tamarin.* [Online]
Available at: https://developer.mozilla.org/en/Tamarin
[Accessed July 2012].

Munzner, T., 2008. *Process and Pitfalls in Writing Information Visualization Research Papers,* s.l.: s.n.

Pietsch, T., 2012. *Trees of life : a visual history of evolution.* 1st ed. s.l.:Baltimore: Johns Hopkins University Press.

Purchase, H., Cohen, R. & James, M., 1996. *Validating graph drawing aesthetics.* s.l., Lecture Notes in Computer Science.

Rendgen, S. & Wiedemann, J., 2012. Information Graphics.

Robertson, G., Czerwinski, M., Fisher, D. & Lee, B., 2009. *Selected Human Factors Issues in Information Visualization,* s.l.: Human Factors and Ergonomics Society.

Rodgers, J. & Lyn, B., 2011. *Exploring Ambient and Artistic Visualization for Residential Energy Use Feedback,* s.l.: Institute of Electrical and Electronics Engineers Computer Society.

Skog, T., 2004. *Activity Wallpaper: Ambient Visualization of Activity Information.* Cambridge, Association for Computing Machinery.

Skog, T., 2006. *Ambient Information Visualization.* [Online]
Available at: http://www.viktoria.se/~tobias/thesis/docs/thesis-final-print.pdf

Sohn, M., Nam, T. & Lee, W., 2009. *The Impact of Unconscious Human Behavior on Sustainable Interaction Design,* s.l.: s.n.

Spence, R., 2001. *Information Visualization.* 1st ed. Harlow: Addison-Wesley.

Stott, J., Rodgers, P., Martínez-Ovando, J. C. & Walker, S., 2007. Automatic Metro Map Layout Using Multicriteria. *Journal of LATEX Class Files,* VI(1).

Sugiyama, K., Tagawa, S. & Toda, M., 1981. Methods for visual understanding of hierarchical system structures. *IEEE Transactions on Systems, Man, and Cybernetics,* SMC-11(2), pp. 109-125.

The Barna Group Limited, 2009. *New Research Explores How Technology Drives Generation Gap.* [Online]
Available at: http://www.barna.org/barna-update/article/14-media/212-new-research-explores-how-technology-drives-generation-gap
[Accessed July 2012].

Tuck, M., 2010. *Gestalt Principles Applied in Design.* [Online]
Available at: http://sixrevisions.com/web_design/gestalt-principles-applied-in-design/
[Accessed August 2012].

Ware, C., 2004. *Information Visualization : Perception for Design.* 1st ed. San Francisco: Morgan Kaufmann Publishers.

World Wide Web Consortium, 2008. *Techniques for WCAG 2.0.* [Online]
Available at: http://www.w3.org/TR/2007/WD-WCAG20-TECHS-20070517/Overview.html#G18
[Accessed August 2012].

Yahoo! UK Limited, n.d. *Documentation API.* [Online]
Available at: http://www.flickr.com/services/api/
[Accessed June 2012].

9. Appendix

In this section you will find all the figures according to the following table:

Please note that all the figures are property of their respective owners.

9.1. Figures

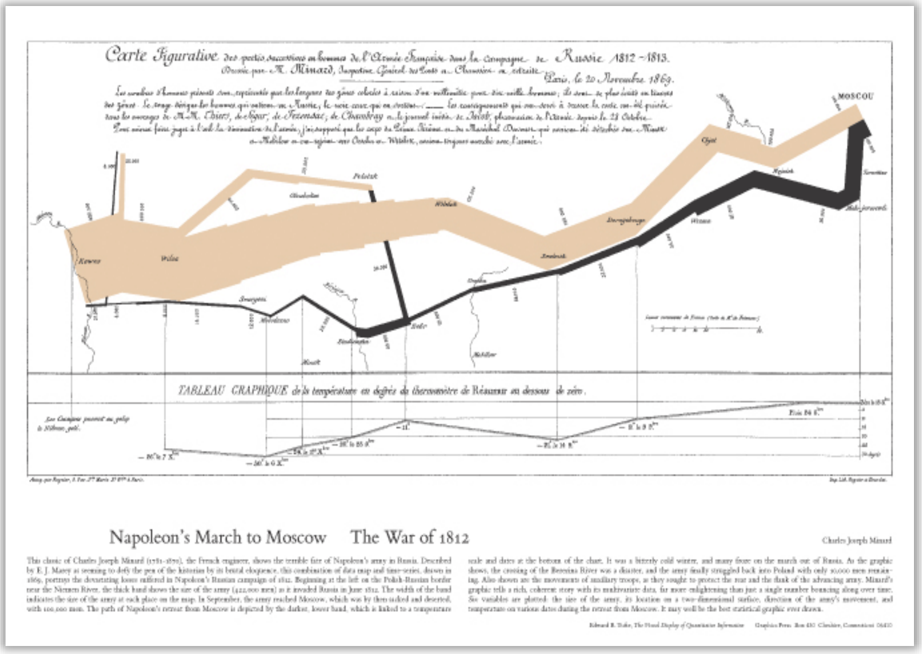


Figure 1 - Napoleon's March in the Russian campaign of 1812

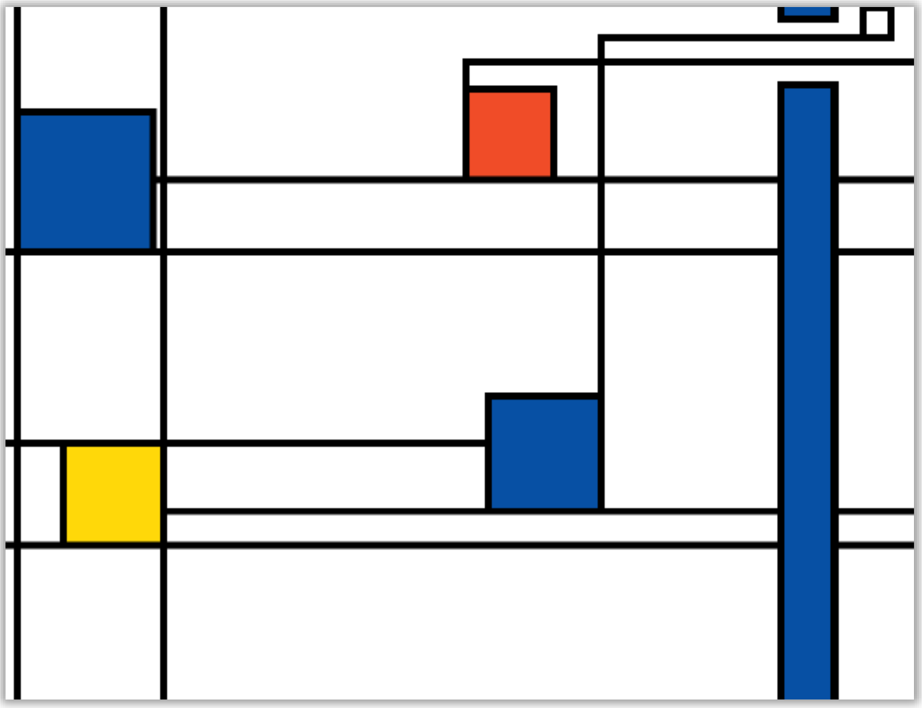


Figure 2 - Mondrian based bus traffic representation.

Blue squares mean the bus is on its way, Yellow the bus is arriving and Red the bus is on the depart. The blue long vertical rectangle represent in this case a river.

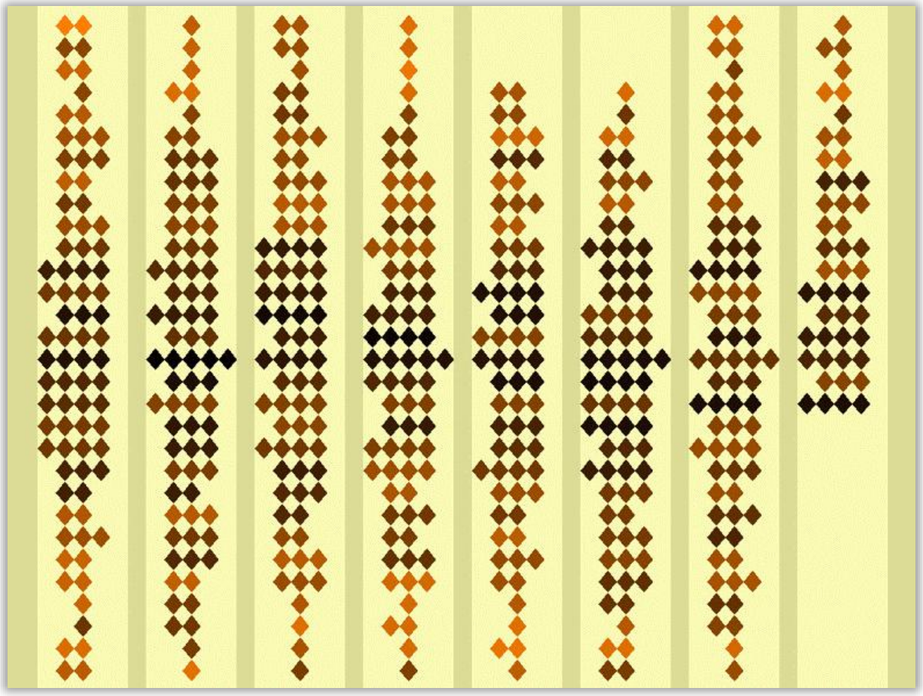


Figure 3 - Activity Wallpaper displaying the intensity of noise in the place

Figure 4 - Andy Warhol based egg-timer visual

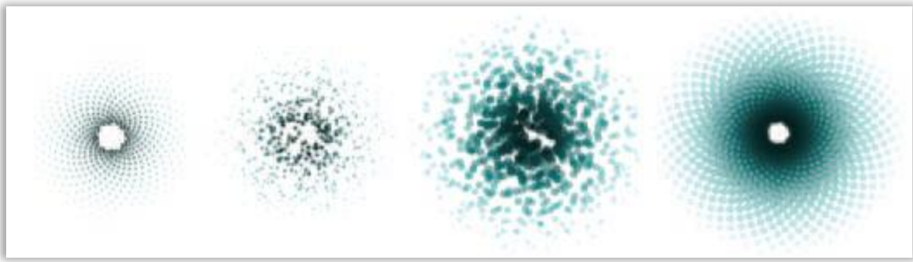


FIGURE 5 - PYLLOTAXIS DESIGN FOR WATER STATE IN ELECTRIC KETTLE.

WHEN THE KETTLE SWITCHES OFF THE SPIRAL RESUMES ITS SYMMETRICAL SHAPE (AT RIGHT), WHILE THE SCALE OF THE DOTS IS MAINTAINED TO SIGNIFY THE ACCUMULATED ENERGY USE.

FIGURE 6 - PINWHEEL DESIGN FOR POWER DRAW

AT LEFT, A FEW PINWHEELS SPINNING LAZILY DURING A LOW POWER STATE DRAW ATTENTION TO VAMPIRE POWER CONSUMPTION ON THE RIGHT.

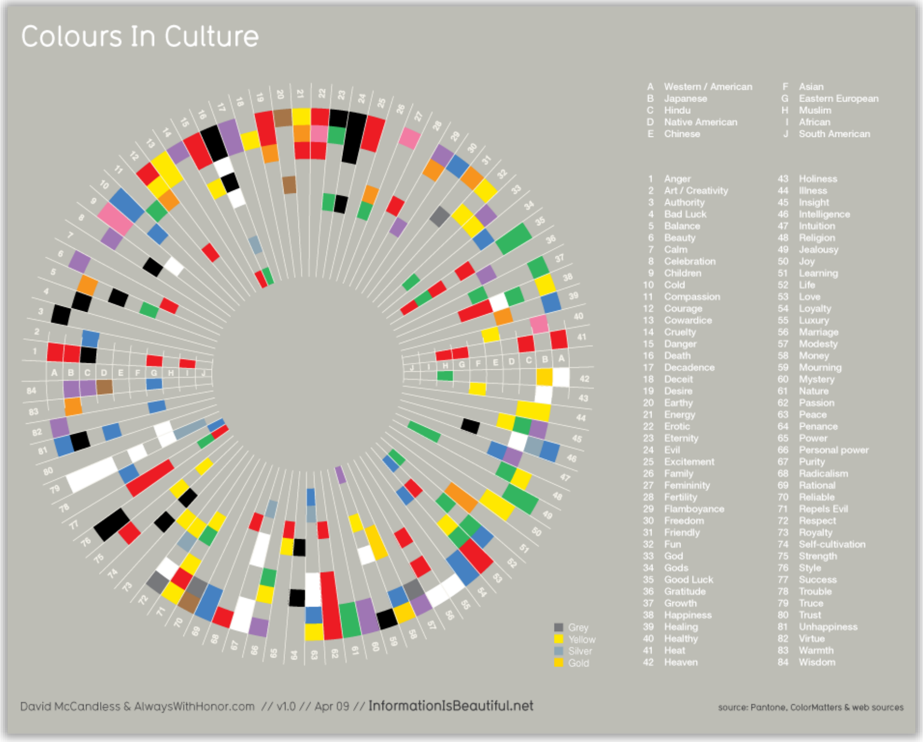


FIGURE 7 - COLOURS IN CULTURE, INFO-GRAPHIC

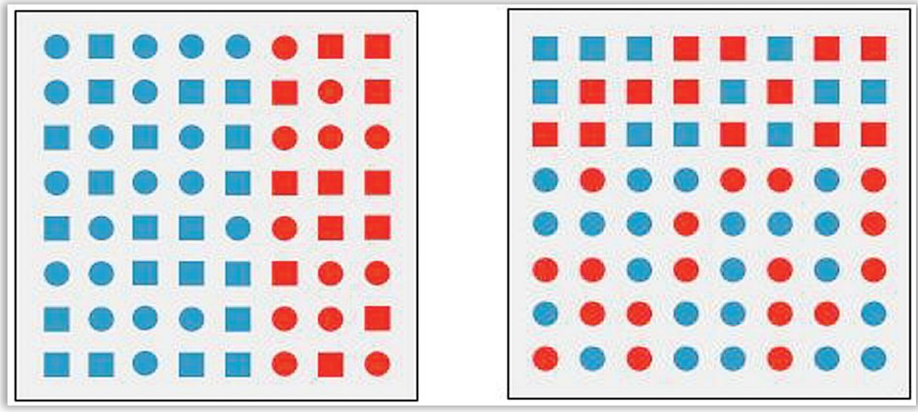


FIGURE 8 - EXAMPLE OF A PRE-ATTENTIVE EFFECT

ON THE LEFT, THE VERTICAL BOUNDARY IS PRE-ATTENTIVELY VISIBLE. ON THE RIGHT THE HORIZONTAL BOUNDARY DOES NOT POP OUT.

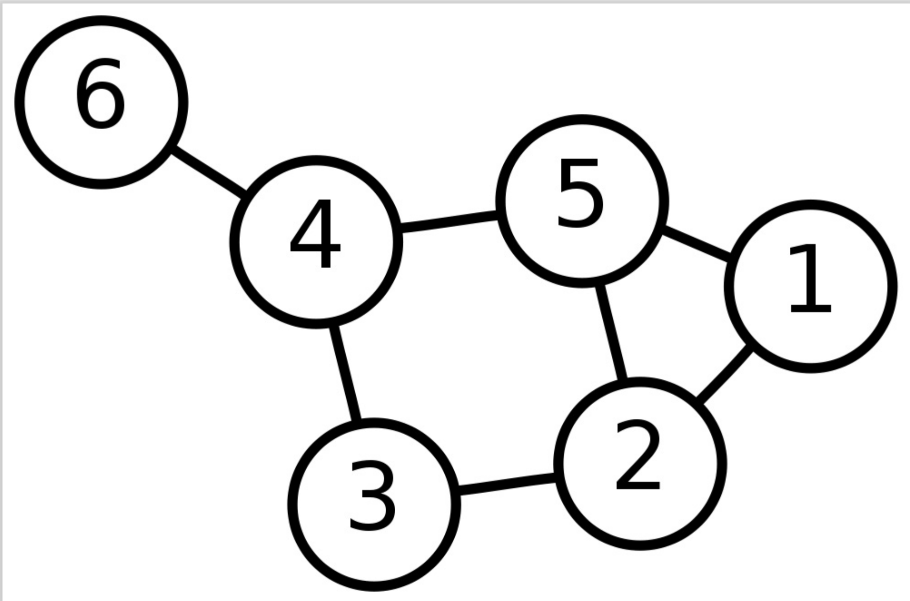


FIGURE 9 - TRADITIONAL REPRESENTATION OF GRAPHS

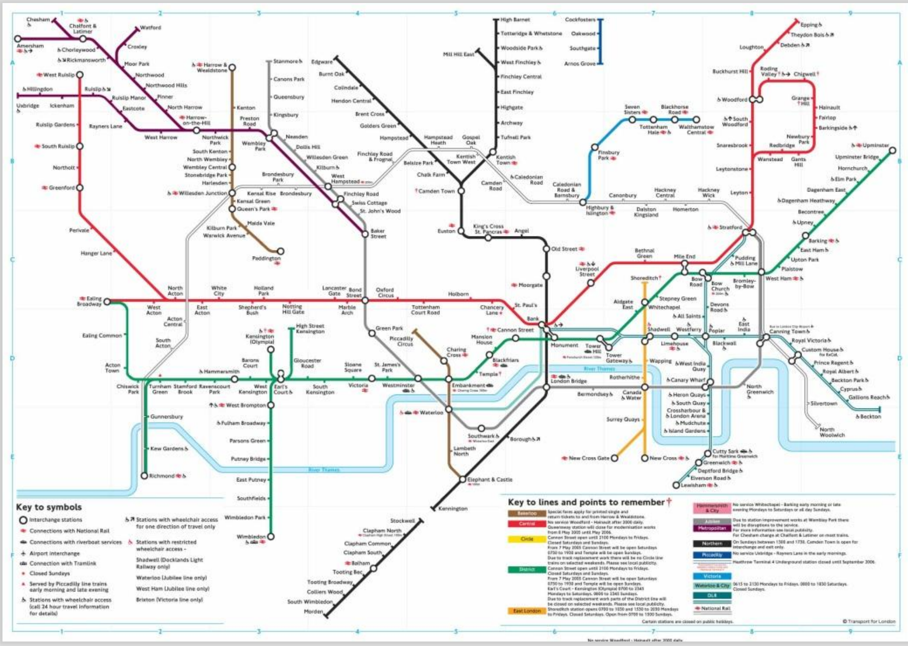


FIGURE 10 - LONDON UNDERGROUND MAP

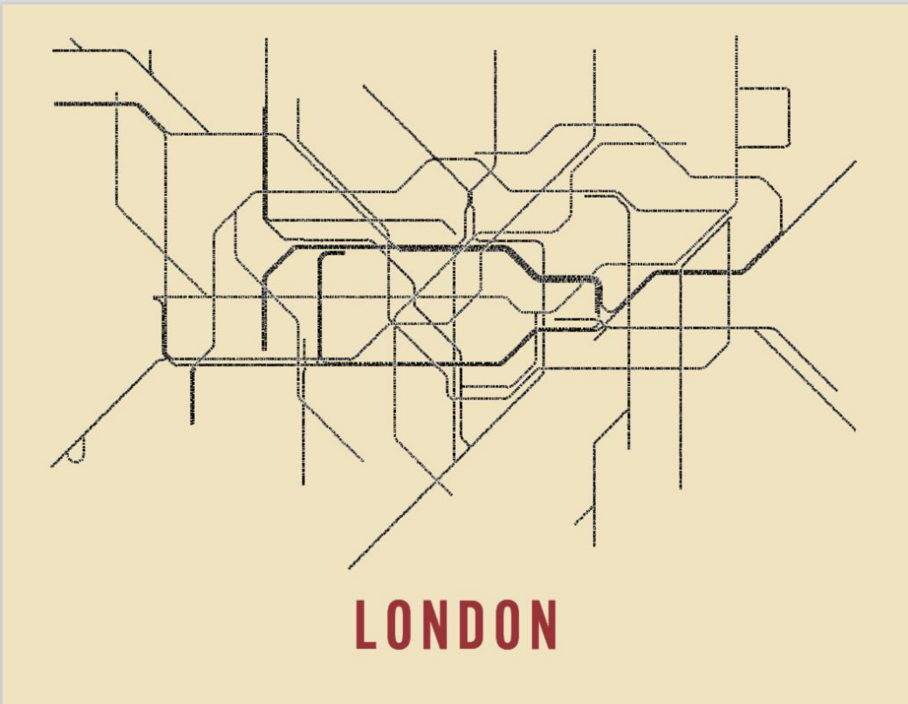


FIGURE 11 - TYPOGRAPHIC WORK ON LONDON TUBE MAP BY FADEOUT DESIGN LIMITED

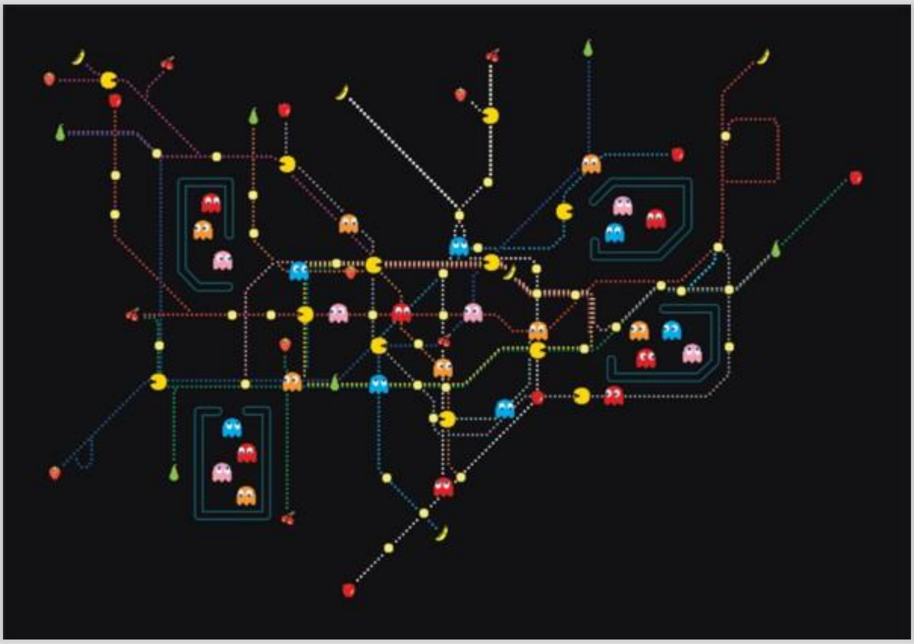


Figure 12 - Pac Man themed version of London Tube map

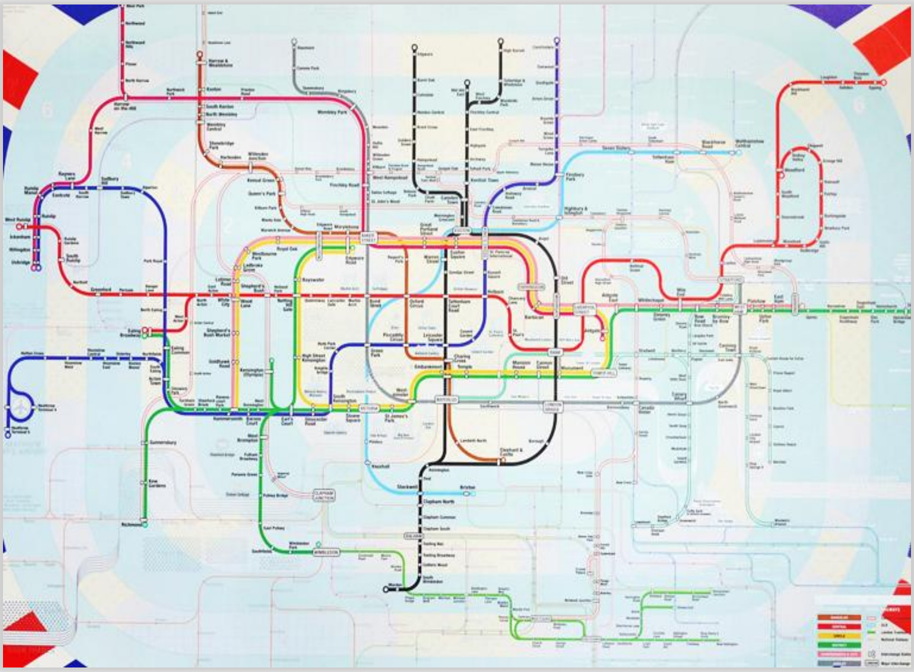


Figure 13 - London Tube map based on zone position constraints

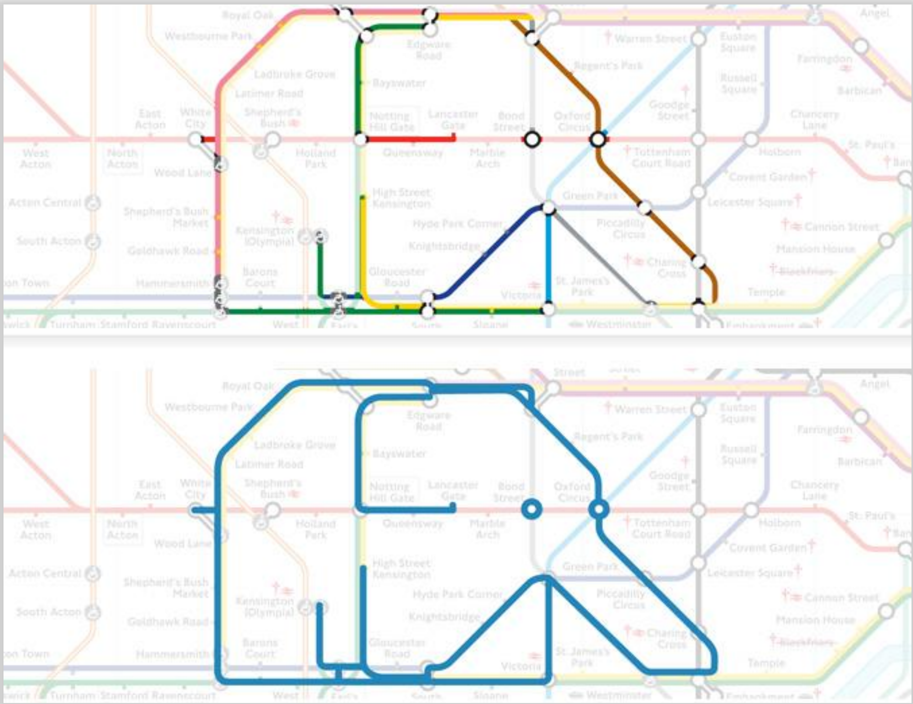


FIGURE 14 - ANIMALS ON THE UNDERGROUND, THE ELEPHANT

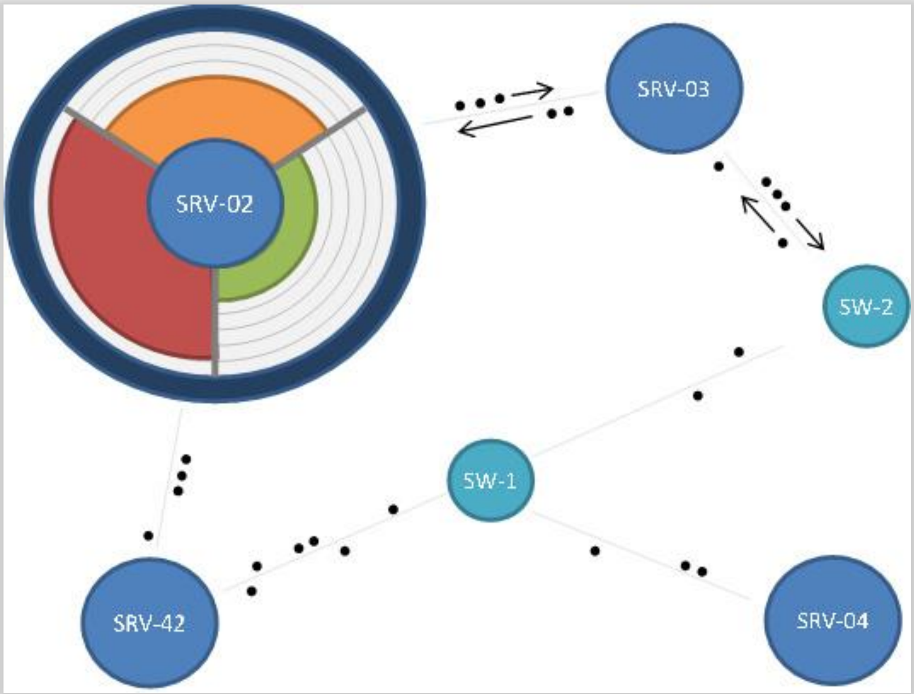


FIGURE 15 - FIRST IDEA AT PROJECT BIRTH

Figure 16 - Stave

We can easily imagine the lines being nodes of a graph and notes the bonds that connect them together.

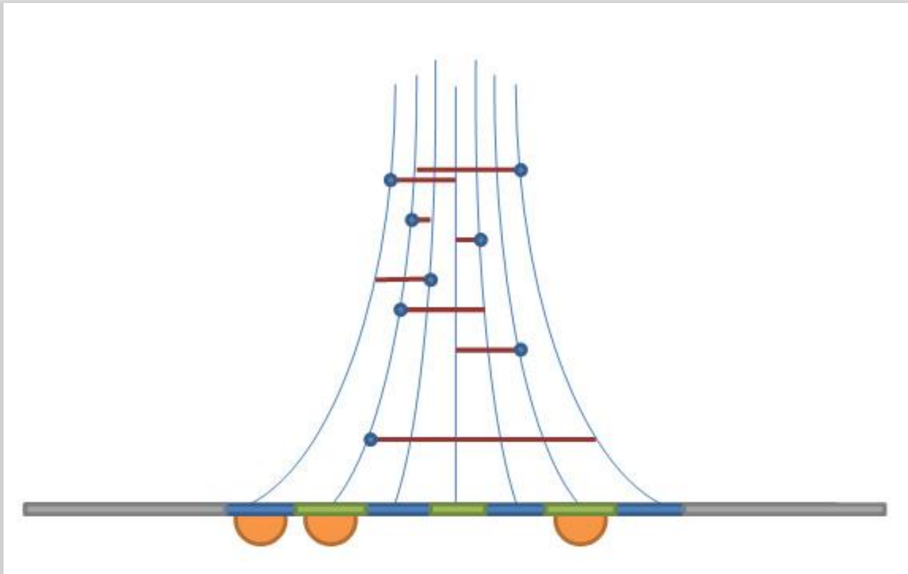


Figure 17 - Second idea based on stave concept

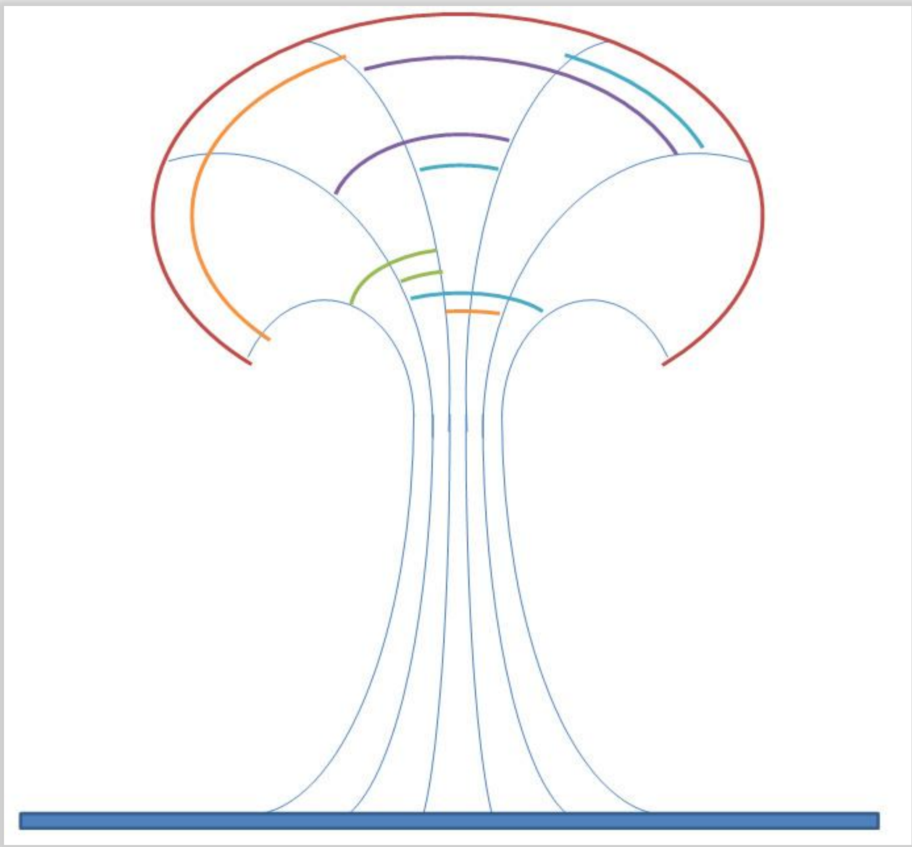


FIGURE 18 - MUSHROOM/TREE SHAPED REPRESENTATION

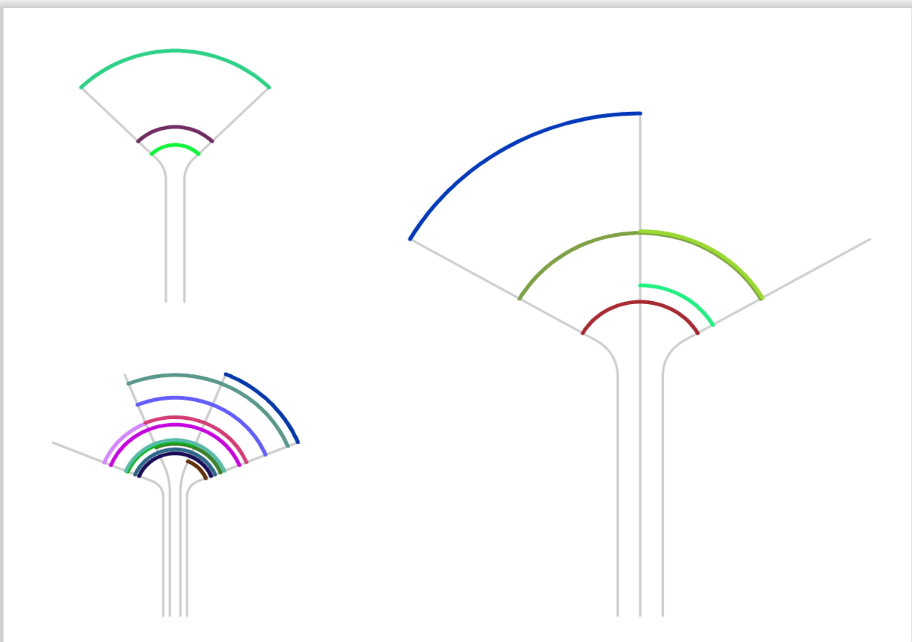


FIGURE 19 - FIRST IMPLEMENTATION OF THE DRAFT

TRAVELS FROM THE 15TH OF JANUARY 2012 TO THE 22ND OF JANUARY 2012 BETWEEN, FROM LEFT TO RIGHT, CANADA, FRANCE, MEXICO AND THE UNITED STATE.

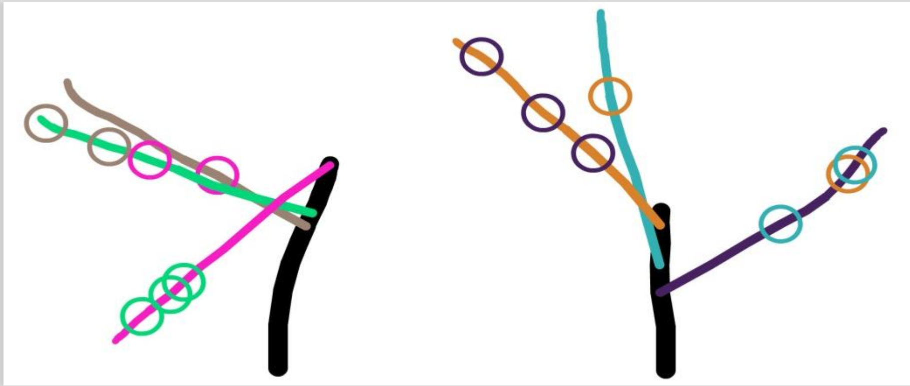


Figure 20 - First step in the "artistic-ness" of our production

We can clearly see the overlapping problem as well as branching growing down.

Figure 21 - Picture showing the considered points for force-based drawing

Figure 22 - Final visualization

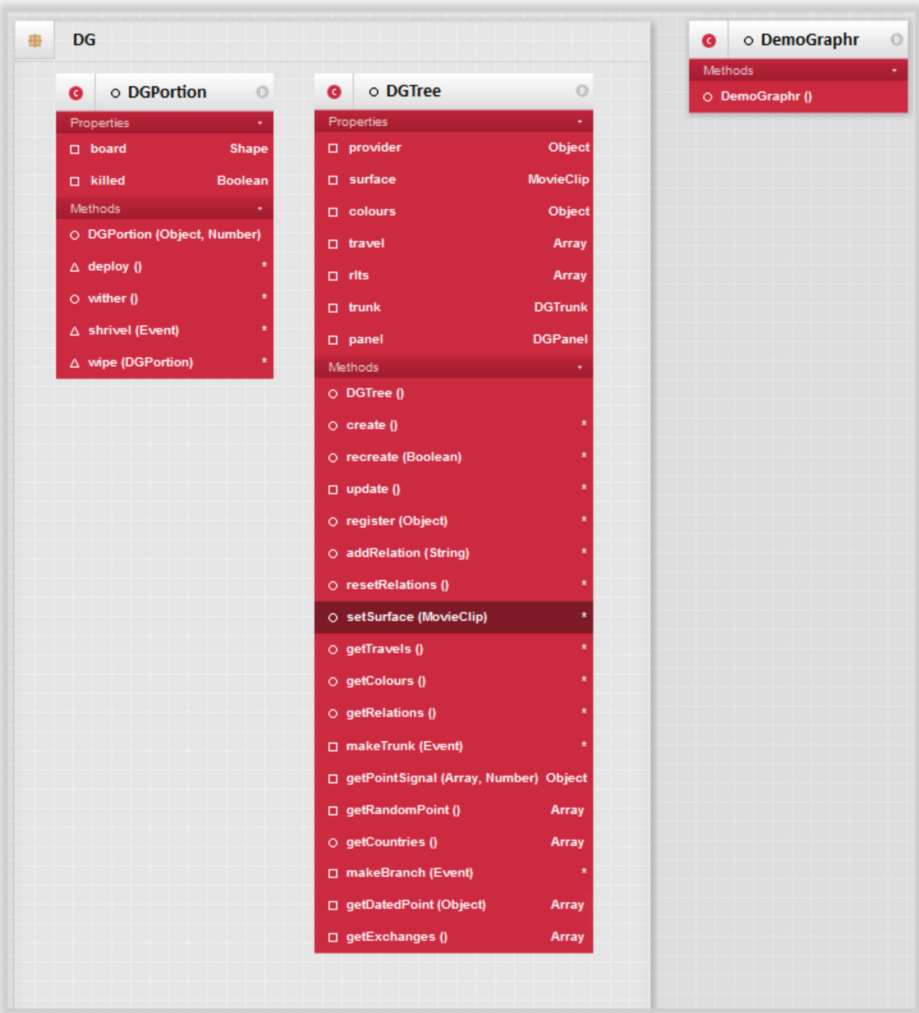


FIGURE 23 - UML CLASS DIAGRAM OF THE PROJECT BEFORE REFACTORING

Figure 24 - UML class diagram of the project after refactoring

(For readability, a PNG copy of this figure has been delivered as a separate content)