Abstract

TODO

Key words: TODO

Résumé

TODO

Acknowledgment

TODO

0.1 Introduction

Graphs can be found in the undergrounds to help travellers find their way, in maps where paths connects various point of interests, in school books to help pupils understand abstract notions. They are a synthesized way to represent information and its usage is widely spread across countries and fields of study. We are so used to them, that our day-to-day life is filled by them without us noticing. This vector of information is not limited by the language, does not require perticular knowledge; it can be used for various context, for various data and can be easily digitilize and thus, transportable or modifyable. Yet its representation is graphical and therefore relies on sight. Finding alternatives in order to access these graphical information is a major issue for the visually impaired population to overcome.

The digital gap is there, and while technology assist everyone's tasks better and faster, people with disabilities are more and more dependent on other's help to benefit from this progress. Braille - which is considered to be the most used form of tactile graphics, is being depreciated for tablets. Although the latter allows enriched interactions and a comforting object to hold on to; they do not combine with the former. The need for a new vector of information is real. More and more data is being produced, this comes with dedicated jobs like Data Analysts and dedicated research like Information visualisation. The increase of research activity in this field has been substential over the last 20 years. We are in the age of Big Data with challenges for a better understanding, visualisation, sharing; yet, accessibility tends to be neglicted.

In the laboratory of the SACHI team in St Andrews - Scotland, a small haptic device has been developed: the Haptic Tabletop Puck (HTP). By moving this special mouse, the user receives haptic feedbacks which can be used for exploration. Although this device allows multiple interactions, the main issue is that there is a single actuator for mapping the height of what is under. The way to explore would be to *bump* onto the limits. If we were to build another device that would allow multiple feedbacks, we would have a more efficient exploration; users would feel where to go. If we were to have longer segment instead of single point under the palms, the users could follow directions. Such a device could be a first window in the accessibility of graphs exploration for visually impaired people as the interaction techniques would be purely tactile. Furtheremore, the audio channel is heavily used for accessibility and developing other interaction contributes in the global autonomy of visually impaired people.

The HaptiQ project aims to design and evaluate an inexpensive haptic device that allows blind users to acquire the representation of graphs and therefore, the key to the understanding of spatial information - like maps, or abstract concepts - like organigrams. The project is framed by Dr Nacenta (St Andrews, UK) and Dr Jouffrais (Toulouse, FR).

I will present in this report the main steps that has led to the creation of this device. I will explain the starting context of the project and the analysing phase; then the design decisions made for this project followed by the way I have implemented them. Finally, I will explain the methods used for the evaluation and an ending discussion about the results and a personal feedback on the skills in human-computer interactions that has been employed for this project.

Chapter 1

Context

1.1 About accessibility

Given the 10th International Statistical Classification of Diseases and Related Health Problems, there are four categories for vision:

- normal vision
- moderate visual impairment
- severe visual impairment
- blindness

Visual impairment includes the last two and from the World Health Organization, it concerns nearly 285 million people worldwide: 39 million of them are blind and 246 million have remaining visual capacities.

90% are in low income settings and 82% are above 50 years old. Being visually impaired is more often something we become and having to learn how to live without sight can be extremely demanding. Only 15% of the visually impaired people know braille. Designing and building software or devices that take into account this need of autonomy, is the real challenge.

In order to build and design for visual impairment, user studies and in-field usage are required. Since visually impaired people are difficult to reach, a partnership with

1.2 Work environments

The majority of the placement has taken place in the laboratory of IRIT in Toulouse - France. It has been the starting point where I have been immersed in a team fully dedicated towards improving accessibility for visual impairment through various project.

A second

1.3 Challenges

Collaboration between two research teams Adapt to multiple skills

Produce something adapted to visual impaired people Produce something that could be easily rebuild Produce something that is cheap

Chapter 2

Analyse

The following chapter presents a background research on the technologies that can be used for graph exploration without the use of sight and a description of the contributions from previous work of the HaptiQ. It will first start with a task analysis on graph exploration. This chapter aims at giving a wider outlook on the alternatives and additional understanding on what characteristics should be considered for graph exploration.

2.1 Task analysis

Finding visually impaired people for collaboration with is not an easy task. The partnership between IJA and ELIPSE tries to overcome this issue, yet many different projects are running at the same time and each one of them needs this worthy collaboration. In order to avoid constant requests of their presence in the laboratories, ELIPSE has the undergoing policy to restrict these requests to the evaluation phase.

This is a major drawback for a user centered development process. Nevertheless, I have managed to find alternative sources of information in order to acknowledge usability issues for visually impaired people.

- A direct contact with Bernard Orniola - one of the permanent members of ELIPSE who happens to be visually impaired. He has provided me with key understanding of this handicap in a day-to-day perspective
- My tutor Christophe Jouffrais, who has a long experience in working with visually impaired people and was able to emphasize some aspects he could feel I was missing
- My participation in general meetings on larger-scale projects such as AccessiMap has given me insight on how to adapt a development process to the

needs of the blind

- A direct contact with X, who was doing his master internship in the cognitive science field. He has pointed out the limits of my design when I have exposed it to him
- Colleagues who have tried a previous uncomplete version of the HaptiQ. They have warned me about the major issues that they have experienced when testing it.

This variety of intermediaries acted as a proxy for infield observatories. Thanks to it, I was able to come up with the following task analysis.

Giving a blind exploration using only a haptic device and a trained user on the interaction techniques, the task would be decomposed into the following steps:

1. Feeling the device
2. Moving (depending on a possible strategy) and waiting for a feedback
3. If there is no tactile feedback, continue moving
4. If there is a tactile feedback, process to understand what the encoded information is
5. Given the new piece of information acquired, adapt the strategy of exploration
6. While exploration is not completed go back to step 2

These steps may seem fairly simple. Yet they give us some clues on the importance of having a recognisable tactile signal during the explroation phase.

Another interesting aspect is the fact that an exploration is the result of a sum of strategies. We can consider that finding the network is the first goal. The next one is to explore all the nodes, which can mean following the network as much as possible. A possible solution to improve this strategy may involve building tactile signals that would naturally suggest strategies and assist the users in their choice.

2.2 Related research

Making an exhaustive taxonomy would be illusionnary as research related to haptics devices has extended its scope and depth over the last twenty five years [ref needed]. This chapter will nevertheless attempt to present technologies used as a way to acquire data through graphs and maps exploration for the sightless.

This background research is based on the doctoral thesis of Thomas_Pietrzak__ on “Dissemination of haptics information in a multimodal environnement” and on the master thesis of Simone__ on “The HaptiQ: A Haptic Device for Graph Exploration by People with Visual Disabilities”.

2.2.1 Braille

Braille is a tactile writing system that has been invented in 1824 and then spreaded around the world since. Although it could be used to read graphs with series of dots, arrows and bullets, it is intended for text reading. The main issue remains the fact that it is difficult to learn. Thinking that all blind people would know it is a common misconception.

2.2.2 ScreenReaders

VIP rely heavily on their audition in order to compensate for their handicap. This usage would even trigger an “obstacle perception” in which they would feel object just by hearing sounds [95]. ScreenReaders provides an efficient alternative to access text and many are available [1]. If only a few screen readers would allow navigation tasks as well, like JAWS or VoiceOver, the main issue remains the usage of audio as a channel for spatial guidance. VIP are not necessarily inclined to use either cardinal points or the four directions (up, down, right, left) to orient themselves. That is why map exploration through a screen reader would require a constant audio feedback. This interaction may provide a useful help for graph exploration, yet it cannot be qualified as the most suited. Besides, it is preferable to interfer with the audio channel as little as possible in order to facilitate the debit of textual information expressed this way. In other words, it would be beneficial to fill this channel with textual and mainly textual information.

2.2.3 Tactile Maps

Tactile maps are made of paper heated to form bumps and relief. It thus creates shapes, lines and dots. These are popular among visually impaired people learning geometry or exploring a map. Even though they offer plenty of tactile freedom, it is easier to grasp a general idea of the shapes by using the ten fingers. They do not provide further interaction unless they are combined with a tabletop such as the Multimodal Interactive Maps (MIMs) project [6]. MIMs is an input output system mixing different technolgies. It keeps the possibility of a ten fingers exploration, but requires a new printing for each visualisation. MIMs fall short

[1] `http://alternativeto.net/tag/screen-reader/` (accessed the 19/08/15)

of rendering VIPs autonomous: scanning and printing would require the help of another person.

2.2.4 Machanical actuators

Presented as the technologic equivalency of braille, they can dynamically change a matrix of actuators in order to provide information which can be a Braille symbol or simple shapes. This matrix can be placed on the finger zone of a mouse like the VTP layer [ref needed] or the Tactiball [ref needed] which implies that the moving hand is also receiving the tactile information or it can be separated like the Tactos device [ref needed] but with a smaller matrix. Their lack of popularity could result from poor quality software applications, as Thomas Pietrzak suggests. Given Jansson study [84] mouses are not compatible with navigation tasks for visually impaired people.

Other displays, like the Brailliant from Humanware [link needed], offer a full range of actuators forming braille letters, but remain fairly expensive.

HTP, a precursor of the HaptiQ, deserves particular attention. One of my tutors – Miguel Nacenta, has been involved in the design of this input/output device with a single actuator in the center [ref needed]. The purpose of the HTP is to explore other possible interactions with tabletops like their further work has suggested [ref needed]. It renders unconventional outputs like friction and softness which can be integrated in various applications. Although innovative, its usage is supported by visual elements and has not been though threw for visually impaired people.

2.2.5 Vibrations

Some devices use vibrations in oder to provide feedbacks. Small vibro-motors can be attached to a glove which makes the device adapted to a hand like the Cybertouch [ref needed]. They could also be integrated on a small surface imitating a matrix of actuators like the Optacon [ref needed].

Vibrations can be used in a matrix of thin vertical panels trigerring a feeling of cavity or bumps when a hand is set on it as in STReSS [ref needed].

Electrovibration is used in the TeslaTouch and Revel systems [ref needed, ref needed]; it is imitating the sensation of friction and is therefore only perceptable when the fingers are in motion.

2.2.6 Forcefeedback

Forcefeedback has known a famous entry in the gaming field with Joystick and Wheels. But their application goes far beyond that. One of the most recurrent names is the PHANToM [ref needed] that forces the point in certain directions. Forcefeedback comes in a variety of techniques in order to push a single point into a certain direction (articulated arm, pantographes, or pneumatics).

Having a single point of contact does not allow users to follow lines easily orto understand shapes [ref needed]. This make Forcefeedback unsuitable for our project.

2.2.7 Air

Feedbacks can be perceived via air motion. It triggers the same signals as with tactile motion thanks to the variety of sensitivy receptors [88, 101]. AIREAL [19] makes this approach possible and uses a motion detector camera as input. Using highly pressured air waves allows long distance interaction (10m). Besides, it is scalable and affordable. Even though they offer a wide range of angles from which the air is pushed, the lack of resolution limits its usage tremendously. Plus, AIREAL is presented as an interaction more suited to enhance user experience than an input output system than for exploration.

2.2.8 No hands involved

(FIGURE: Homonculus sensoriel)

If we were to represent the human body by its touch sensitivy, we would end up with a weak figurine with enormous hands, lips and tongue. This is maybe why bolder interactions are exploiting the latter with the Tongue Display Unit [9]. This display places a seven by seven grid filled with electrodes on the tongue and can be used in a no-hands-involved scenario: as for instance a working surgeon. Others would use the brow with the Forhead Retina System [ref needed].

Although intriguing, both of these displays allow limited interaction and are suited for very particular scenarios.

2.3 Previous versions

FIGURE haptiQ evolution, tactons

In 2014, Constan Simone has worked on a first version of HaptiQ at the University of St Andrews. His development process was focused on the engineering of a device to handle multiple actuators. These actuators could therefore have their own language in order to transmit information. He has designed multiple cases for the HaptiQ and maintaining all the servomotors.

His work on a background research narrowed the disadvantages of other haptics solutions. He has also implied that a vector based mechanical actuator such as the HaptiQ is unique. His ideas on possible applications in order to help math signal representation (like in Figure ?) are highly valuable.

Even though his design on the caps does not appear in his report, we have to give him credit for it. His work on tactons seems promising; yet, it is not backed up by any user study. This imposes its reconsideration.

Simone has also managed to extend this first version with button and has started to work on different possible interactions with pressure.

Finally, he has briefly pointed out the issue of having multiple wires running in order to control the servomotors which has led me to prefer solutions allowing the device to be as nomad as possible.

2.4 Conclusion

Haptics devices demand material and often electronic circuits to be build. This results in high costs overall and is often dedicated to a specific usage. If our goal is to provide a solution for VIP around the world, then we should take into account other aspects such as making it easily replicable and allowing applications to be build on top of the key interactions like the Haptic Puck Tabletop and the Phantom did. But this goal requires various skills and a careful design.

Many alternatives exist, but the same issue remains: we are too focused on data represenation than on data meaning. It might be more relevant to focus on the general trends than on the exact measurements. Let us remember that it is really hard to learn the simple concept of a squared angle when one would be deprived of sight. The challenge is there: trying to give a natural interaction for the strategies involved in exploration and facilitate learning. A way of solving it is to take a step back in the representation of information: we are not interested in the value of a perticular pixel but its meaning, its purpose. Is it a part of an edge? Is it filling a cue point? Or is it just random noise? These problems can be solved by giving meaning a perticular point; this is why we are focusing only on graphs. They are a scalable and precise representation of the key information. Understanding graphs is mastering a way to easily acquire conceptual and spatial information. (ref: 01__)

Chapter 3

Design

Tactile vector based devices are inexplored in haptics research, our aim is to provide some feedback on their usage for graph exploration. Yet, the key points are to design a device that is affordable and easily reproducible. Our design path also expects to reach the open-source community in order to ease further contributions. In this chapter, I will present the device and its functionalities and discuss the different choices that have been made in order to build a functional prototype for our research.

3.1 Global design

The HaptiQ can be seen as a movable device with tactons to aid graph exploration. These tactile feedback are given by actuators on the top of the device forming a star shape. They are directly linked to the hands. Moving the device is an input data provided by the user, just as if we were to move a mouse. This motion is then tracked and processed by a software layer that will trigger a certain tactile signal, called tacton. The device will lif the actuators up and down according to this tacton signal.

This is how the HaptiQ itself works and the software must meet its requirements. We can also go beyond these first requirements and provide a standard API for interaction to be integrated into the software allowing one interaction to be loaded from a list. Such a software Although this is the main workflow for the HaptiQ, we do want to be able to compare this interaction to other techniques. This is why special efforts have been made for the software part that I will detail.

The feelings that the users experienced by using this device – the tactons – have a direct impact on the usability of the whole system. Human Computer Interactions (HCI) skills have been applied in order to make sure that the final tactons are suited for the task of graph exploration.

Although important, these tactons are supported and limited by the hardware. It has required various skills in order to build the case, assemble the electronic components and set up every actuator.

Figure x_ represents the glogal structure of the HaptiQ. We have a first agen taking care of tracking, the software, a tacton

3.2 Design of the hardware

The HaptiQ receives the tacton signal to execute. The tactile sensation is coming from a rubbery cap that is being vertically moved by a servomotor controlled by an Arduino electronic card. These components are placed inside a 3D printed case. This chapter will detail each part of this hardware.

3.2.1 Actuator element

An actuator element is made of the following parts:

- a cap with a rubbery feeling
- a vertical plastic stick that supports the cap
- a servormotor that transmits a vertical motion to the plastic stick
- a 3D printed servo-holder which offers an appropriate casing for the servomotor

All these elements were brought up by the collaboration of Conte Simone and Hoggan Eve. who have previously worked on a first version of the HaptiQ. The cap is made from a special material that can be used by a 3D printer and this gives a soft, yet elastic feeling. The shape can be described as a segment with a height on a top of a triangle. The vertical plastic stick enables to move this cap above the servomotor and is fixed to it by a small rubber. The vertical servomotor are one of the best ratio of small and inexpensive - they cost each €12 and are about 2cm in height. The servo-holder is a design made by Eve.

The actuators were hold but not nicely maintened which would lead to some pressure to the wires and an uneven tactile sensation on the hands. From my point of view, solutions for hardware problems need to be revertible as I were dealing with specific servomotors and 3D printed objects that can take a lot of time to reprint. I have first tried a very basic solution of using bluetack as a way of soft maintener which did offer a short run solution. It was still too untight to be practical. So I have drilled the servo-holder in order to make a even height position for all the servomotors and leaving some space for the wires to go through. An elastic rubber band was then placed to avoid jiggling movements from them.

In order to be sure the servo holder stick well onto the first layer of the case my best option was scratch. It does not damage the objects, it is not messy like the glue and it allows adjusting the servo-holder under a minute.

(figure of a drilled servo holder)

In order to work, the servomotor needs to be powered and controlled by an electronic board.

3.2.2 Electronic components

Arduino is an electronic card that is backed by a huge opensource community. This makes the workflow of running programs on it fairly easy and highly documented. Because of its flexibility, many other electronic firms have built shield or extension components to enhance the possibilities or the card. This is the case of the Adafruit card that we are using on top of a Arduino Uno. This extra shield allows to easily map the circuits of the servomotors to the Arduino card which enables their control in the programme.

In order to make the device nomad, two batteries are needed one for the Arduino, one for the Adafruit. Yet, the commands could not be received, which has led us to add a bluetooth component and turn the HaptiQ into a fully wireless device.

3.2.3 Case

The case is also a design provided by Eve H as she has preivously worked on another version of the HaptiQ. Having it this massive did lead to some concerns, but it is actually more impressive than a problem. Ideally,

Besides, even though 3D printing is widely spread - the actual process of going from zero knowledge to the printing of such pieces can be and was time costly. For instance a poorly configured printer took about 30 hours to print all the mention parts for this device.

Although a new promising design was delivered by Eve, the timing was too short for a risk free transfer.

3.3 Design of the software

The software is the agent that enables interactions for graph exploration. It has been made flexible enough to change the used set of tactons on demand. The architecture has also been designed in order to facilitate a change of interaction technique, again on demand.

We need a way to evaluate the HaptiQ in order to understand its assets and failures in a task of exploration. One way to achieve this is by comparing several interaction techniques for the exact same set up. This puts a requirement on the software to be able to change interaction easily keeping the exact same set-up.

The final and current state of the software has been achieved by an iterative exploratory approach and many refactoring. This has led to three main threads (Input listener, Graphic User Interface or GUI, Interaction executer) communicating through four main components (Network, Interaction, Device).

3.3.1 Graph component

This component is used as the container of all the geometric logic that are involved in graph exploration. More precisely, it contains the collection of Nodes and Links that respectively inherit the functions for Points and Lines. These collections can be processed by the GUI and appear on the screen. An existing libraries that provides functions for graphs and network manipulation was considered [1]. Our needs were too perticular in order to use such an extern ressource easily and on the same time, the computation complexity for our need remained sufficiently low to be easily done during this internship.

(figure showing up a graph)

3.3.2 Interaction component

A graph exploration requires two things, what is explored - the network, and how -the interaction. This component has came late in the development process as the first objective was to make the HaptiQ work and then enable multiple interaction techniques: which requires more thinking and a better outlook. The interaction component makes sure that all the derives interactions follows these three basics functionalities:

- open: which will verify if the system meets the requirements for this interaction and therefore avoid any error

[1] `https://networkx.github.io/`

- process: is called by the interaction thread every loop turn; it computes the appropriate output from a given situation and executes it
- closed: which would close the remaining process linked to this interction opened during its usage

By such a structure, it was easy to adapt the HaptiQ interaction previously made and at the same time offering a standard way of creating an interaction technique. Besides, it offers a control over the execution time which is necessary for our user study.

During open, this component makes sure that a network is available. For more specific interaction like the HaptiQ, it checks the availability of the device component. This is how the input is made available during process for computation.

Building this interface has turned a first restricted version of the software into an evolutive program that can now accept various interaction techniques. Future collaboration has been made available by this refactoring.

3.3.3 Device component

The device component is a virtual representation of the HaptiQ. It has therefore a position, an orientation and the state of each of the actuators. Each actuator is characterized by an angle or a direction - like North which would be equivalent to 90°, that is fixed and a level - between 0 and 100, that can be changed.

As it is a representation of the device, all the interaction techniques that are dealing with the HaptiQ are using this component as the reference for the position. More than a representation of the current state, it is actually the state in which the HaptiQ should be. Attribute like the position are directly depending on the user and can only be updated, but it is a security regarding the levels of the actuators. It happens that the HaptiQ does not execute the latest tactons and gets stucked in an other state which does not match the real situation. Using this device component as the reference for the level of each actuator makes the system less prone to context errors.

3.3.4 Input tracker

This is the thread that is constantly listening to the information regarding the HaptiQ. These information are shared by the TUIO protocol. It simply means that the information is formatted in certain rules which makes the parsing process easier. The tracker receives a variety of data in the form of chains of characters. Because they follow some patterns, it is possible to extract the key information which consists of a variety of points. These points are then parsed to a handler that compute them in order to obtain a center point and an orientation. The position and the orientation of the device component are then updated.

Basing the position of our device on the computation of data sent with a high debit does lead to some incoherences. This issue has been solved by adding on the tracker a checker that allows only valid position.

3.3.5 Graphical User Interface

The GUI or simply called *view* in the program, represents the network loaded visually as seen on figure x. It serves as the reference for which network and which interaction are being currently in use. A new network can be loaded on live, this allows future applications of the HaptiQ in which a user could change it himself.

It also includes a special window that acts as a visual representation of the tactons currently in use for the device. This has been extensively useful during the development phase in which sets of tactons were visually tried before any further development. This could be seen as the low fidelity prototype format of tactile interactions as it gives a genreal hint of how the tacton will react; yet, the lack of sensation makes it a very low fidelity.

3.3.6 Interaction processor

This thread checks which interaction is selected by the view and will call the *process* method for that interaction. For each time the interaction is changed, this processor will make sure the previous one gets closed properly and the new one *open* - as described in the interaction component.

3.4 Design of the tactons

For our device, a tacton is the position of all the actuators for a given time or for a short lapse of time. This time would be the evolution of the levels until they repeat the pattern - like an oscillation. The tacton is the language in which we are communicating what is drawn under the pointing device. It could be a node, a link or nothing at all - but each one of these situation leads to completely different tactil signals and needs to be easily recognisable. One of the goal of the internship is to evaluate the usability of each tactons.

In order to establish the most suited sets of tactons, I have proceeded by iteration. For each actuator we have a range of height - which allow the creation of interesting patterns like oscillations. Now, with eight actuators, the possibilities skyrocks. Proceeding by iteration helps to narrow the needed characteristics and avoid wasting development time. I will explain in the following section the three main stages that have guided me towards the current version which is still under testing.

In order to understand how tactons work, we have to explain for which contexts they apply. Here are the following states that are considered for our tactons:

(figure on node)

(figure on link)

(figure near)

(figure on nothing)

3.4.1 First iteration of tactons

My first iteration has proven me the good distinction between static and oscillations and that an actuator should guide one graph element. I also had to find a way of rapid prototyping a usual 'paper design' is not possible with tactile signals.

It has been evaluated by walkthrough on a visual feedback the 8th of April 2015 [2]. I have decided to represent visually the state of each actuator - the darker, the higher. This has resulted in a relatively low fidelity prototype of my set of tactons, yet it is sufficient to observe basic usability problems. I have had to find my alternative of low fidelity prototyping and this is how the visual representation came up.

Because of the early version of this interaction, links were not integrated yet. The tactons to be generated depend on the following rules:

- near a node, the tacton indicates the closest nodes by up and down oscillations. Actuators moved this way are the two closest angles, so if the node is at 40°, the North (90°) and the East (0°) actuators gets moved.
- on a node, the tacton indicates the closest nodes by being fully up. The concerned actuators are the same as previously.

The intention in this set of tactons was to encode as much information as possible. By using this perticular set of tactons, one would know when he would be near a node because the oscillations would begin; at the same time you would

[2] `https://github.com/asiegfried/vegham/tree/v0.1/app`

still know about nearby nodes. You could easily distinguish when you are on a node

That was in theory, while experimenting roughly with my low fidelity feedback, the subjects were feeling lost during the whole exploration process inspite of me showing where were the ndoes. The following interviews have revealed the reasons:

- there was far too much information at a giving time
- the interactions felt unatural
- it was impossible to tell how many nodes were nearby

Although this interaction was highly depreciated, the task of know whetether or not we were on a node or not was done accurately. A first contribution from this first iteration is the efficient distinction provided by static versus oscillation. This characteristic has been preserved through the versions. A second one would be the fact that having more than one actuator guiding a single node was too complex too be easily processed by the user. This aspect has been taken into account in the next iterations.

3.4.2 Second iteration of tactons

The second iteration has brought me consider simple tactons first and high contrast could be one of the most important characteristic I was looking for.

Another path was explored from my first iteration, I have tried to find simpler tactons that could still provide guidance [3]. The following rules conern this second iteration:

- near a node, the tacton indicates the direction towards it with a certain intensity. Only one actuator is moved for this tacton, it is the one closer to the angle. For instance for a node at 40° it will be just the East (0°) actuator. The intensity is inversly proportional to the distance. The closer, the higher the level would be.
- on a node, all the actuators are higher than normal.

This interaction takes into account what has been remarked in the first iteration. One actuator is for one node. Oscillation were reserved purposely for the links, that have not been integrated to the software at that time.

[3] `https://github.com/asiegfried/vegham/tree/v0.2/app`

Another walkthrough has been tried on this interaction in order to detect usability errors and just in general seeking other ideas. This interaction has received several positive feedbacks. The sudden change for when the pointer is on node makes the message very clear. The growing intensity also indicated well the exploration. The major issue remained the fact that these tests were based on visualisation as a proxy of what the tactile sensations would be. Obviously these two senses cannot be considered equivalent for my tactons; I had then reached a limit for my low fidelity prototyping.

Yet, I have understood that simpler is generally better when it comes to provide guidance. This aspect has motivated my further interaction. The major contribution of this iteration has been the importance to keep a clear contrast between the two situations: on node and not on a node. Since the major difficulty is to find the network, it must be very clear for the user when it is over a node or not. It accentuates a mental marker on that very specific zone, it is also reasuring to have such a clear and distinct tactile feedback.

3.4.3 Final generation of tactons

A few other tactons have been developed while waiting the HatpiQ to be build. After some hardware issues (that will be presented in Implementation), I was able to provide the real sensation of the HaptiQ and this was highly valuable in order to seek the features that would lead to a suitable tactons.

After several tries through the hardware capacities and my self judgment, I came up with a last generation of four sets of tactons. The goal was to compare them in a user study and being able to justify the most appropriate one for graph exploration. During the first tries out of this user study, I had to withdraw two of them as they appeared to be completely unusable for the required task. Two of my collaborators, one visually impaired one not have experienced the same struggles in using some of these tacton sets. Among other issues, the users felt overwhelmed with the tactile information - like arriving on a node, all the actuators were moving at the same time. And also, it appeared that the intensity that felt like an interesting idea in the second iteration, turned out to be completely unperceptable in the real situation. We can sum up that the main reason why they were not efficient is because of their lack of simplicity and consistency. I had to remove them in order to focus on the most promising ones.

The two remaining tacton sets are the result of an iterative exploratory and are to be compared in a usability study. One can be considered as a direct mapping of the underneath situation when the second provides an additional guidance.

Mapping

This tacton sets simply encodes into tactile feedbacks what is directly underneath the device. It has been narrowed to three very strict rules:

- on a node, the actuators which direction corresponds to the direction of a connected node are up, the rest are down.
- on a link, the actuators which direction are parallel to the direction of the link are oscillating up and down on an high level, the rest are left down.
- on nothing, all the actuators are down.

When moving the device onto a node, some actuators goes from a down level to a up level: their is a high contrast between these two tactile situations which respects the criteria of a high contrast found during the second iteration. We have also made good usage of the duality of static versus oscillation as they both encode distinct facets of the exploration. Static is for the nodes and emphasize on pausing and maybe remembering this perticular point. Whereas, oscillations are for travelling between nodes and this constant feedback of the direction to go can be seen as an encouragment to proceed.

Guidance

Very close to the previous set of tactons, Guidance offers just one more rule in order to help keeping track of the network.

- on a node, the actuators which direction corresponds to the direction of a connected node are up, the rest are down.
- on a link, the actuators which direction are parallel to the direction of the link are oscillating up and down, the rest are left down.
- near a link, the one actuator which direction is the closest to which the link is, oscillates in a low level.
- on nothing and near nothing, all the actuators are down.

Just as the Mapping set, this one respects the criteria established during the two previous iterations: high contrast and static versus oscillation for two different exploration phases. It includes a quick guidance that helps user to return quickly on their track. Even though a new tacton is used, the help provided can be worth it. The questions rised by this alternative are untangled in the Evaluation chapter.

3.4.4 Remarks

I have not talked about a basic criteria which is to prevent a single tacton signal representing two distinct situations. It is the first level towards consistency, obviously. As one would notice, the sets have been constantly moving towards simplicity and contrast. One can argue that providing guidance is obviously more usable, but since the beginning of my internship I have been surprised by the difficulty of finding the key elements for a good tactile sensation. I have not taken this for granted and this is why I felt the user study is justified. Besides, providing some analyse feedbacks on the differences of mapping versus guidance can surely be seen as a minor contribution in the understanding of tactile feedback based on vector for graph exploration. We may appreciate the fact that, as an engineer it is easy to see many different ways to encode in tactons the underneath situation of a pointing device. As challenging as it seems, this approach does not consider the usability aspect.

This chapter has described the design of the software, the tactons and the hardware. Furthermore, it has detailed the reason of the iteration over some of them. We end up having a relatively inexpensive device - around €300 and reproductible. The software is opensource and using the HaptiQ interactions is cross-platform; it is even designed to welcome new interaction techniques for the device or to ease comparison. The implementation of this design has lead to some rationale desicions as well which will be detail in the Implementation chapter.

Chapter 4

Implementation

Technical disclaimer

4.1 Tracker

What enables the device to be tracked is a frame from ZaagTechnology that sends through a TUIO protocol the information of all the points tracked inside the frame.

Although this system is currently only working with the ZaagTechnology frame, the TUIO protocol makes it completely independent of the tracker used. For instance, before receiving the frame I was able to simulate the same behavior with my phone and the

4.2 Software

git (stats?), python, POO, testing, refactoring, rst for docs, PEP and Python, how the tracker works, usage of the feedback windows because of the dual here/there, cross-platform

4.3 Hardward

Arduino, connecting wires, 3D printing, bluetooth vs Wifi, possible enhancement with buzzer / piezo, device component sends a lot of infos but we only deal if it's a change,

? why TUIO

4.4 Interactions

HaptiQ, Guide, Keyboard

Chapter 5

Evaluation

The purpose of this internship is to be able to understand what makes a good tacton sets and how does it compete with other interatcions already used for exploration. To answer those two questions we have conducted two user studies.

Beyond simply gathering data on the usage of tacton sets, we are incapable of chosing a set over another; therefore, our first user study is a justfication of a choice used in the second one. We can claim that since that specific tactons set is preferred, it is the most suited competitor against other interaction techniques.

5.1 User study A: about the HaptiQ

Two sets of tactile signals have been developed: one is purely mapping what is under the device (M) when the second adds some guidance (G). These two sets - M and G, are the result of an iterative process where we were able to improve the tactons as described in the Design section. As described in the Design section, four sets were initially prepared for this user study, but two of them has proven very little usability in the first tries.

5.1.1 Hypothesis

We are looking for an answer on the best set of tactile signals, in order to prove or disprove the usability of one we will start with the following statement. With the end results, we would then be able to confirm or deny our hypothesis.

1. The set G is more **efficient** than M.
2. The set G is more **effective** than M.

3. The set G is more **satisfying** than M.

5.1.2 Procedure

The subjects were informed about the HaptiQ project and the purpose of this evaluation. They were given a disclaimer to read and sign before starting the experiment. A form was then given to fill out - this form was to check if there is any training for the subject on haptic devices. Before the eye mask was placed, they were told that they could leave the experiment at any given time without having to justify.

They were then instructed to manipulate the device in order to feel it without tactile signal. This is a way to make sure the users were comfortable holding the device and the time frame was also used to answer questions regarding its holding or its robustness. Besides, as mentioned in our task analysis, a first step for exploring a graph using a haptic device is to get familiar with the tactile sensation. Ultimately, this free ride helped the subjetcs getting a mental representation of the frame that represented their workspace. I emphasized on the importance that it was an evaluation about the interactions and not their personal performance. Another crucial point was to give a purpose for their performance which was speed - this has been shared the same way among all the subjects. When done, a training network was loaded with one of the two sets. I then described all the tactons involved in this set and the way they are triggered before letting the exploration begin on the training network. I purposely asked them what they felt and if they were understanding the meanings of the tactons. I was also assisting their exploration in order to avoid frustration. When ready, I loaded a second graph that was used as a blank test; we could then agree on a way to share the answer eg. "a central node with one connection to the North and one to the South-East". I then told them that there would be six similar tasks to perform as fast as possible and that I would give them a "Go" which meant that they could start anytime from the signal as I have started the recordings manually on their first move. I believe that this procedure helped to relieve some stress for the start periods.

The graphs used were always a central node in the center of the frame with one, two or three connections around it. This information has been shared among all the subjects before begining the evaluation. The application allows generating such type of graphs randomly depending on which number of connections wanted. For the six tests, a random order of the following possibilities was selected differently for each subject:

- one graph with one connection
- three graphs with two connections

- two graphs with three connections

Other graphs were planned and we were to retry the same experiment as explained in [ref needed], but the first tries have revealed strong difficulties in their exploration. Although possible, we have preferred to go with smaller and simpler network giving us more time for more tasks to be performed.

When an interaction had been evaluated, a SUS questionnaire about satisfaction was then given to the experimentator. Remarks on that specifc set of tactons were taken into account after it being filled. When done, the process was tried again with the remaining interaction. At the end, the subjects were asked for a preference and remarks on the device were then co

5.1.3 Measurements

For each task, the following data were computed:

- quality of answer (Wrong or False)
- time spent from moving the device to hands off the device - before the answer was given
- ratio of the time spent on the netwrok out of total time of exploration
- distance travelled

We could therefore regroup these measurements into the three main characteristics of usability:

1. Effectiveness: quality of the answers
2. Efficiency: time, ratio and distance
3. Satisfaction: SUS questionnaire and the remarks

5.1.4 Subjects

All the subjects were between 24 and 27 years old, half male, half female. Two of them were left handed, yet they both used the mouse on the right hand (they were not asked to do so, they just did). None of the subjects had a regular usage of haptic devices, although some declared having little to moderate experience with haptics in general.

5.1.5 Results

All the subjects has preferred the guidance tactons and this is backed with the SUS scores which have all been higher for G than for M. The T test of Student for paired values gives a highly significative p-value (0.004693) which suggest a real difference.

Six errors have been made with M and five with G; given that small differences we are in no position to confirm nor deny our hypothesis on a more effective set of tactons.

On a qualitative level, several remarks have been made by the participants. They have shared a feeling of being lost or having a broken device when exploring too long outisde of the network with M. The noise has played a part in their ability to understand the situation. Although we might see this as an interferance with the usage of the HaptiQ, we can accept a reasonable amount of noise when using the device as it may help to get whether or not there is a sudden change of tactons.

We have several indicators going towards a clear preference for the G which conforts our intuition. Still, the lack of effectiveness raises questions. subjects, results (show some graphs), explain about parsing

5.2 User study B: about graph exploration

The user study is still in try-out as this report is written: the current state

About interaction technique

Hypothesis

5.3 Hypothesis

5.4 Protocols desgin

5.5 Results

Chapter 6

Discussion

6.1 About the project

6.2 Acquired skills

6.3 Remarks

This report has been generated with RsT, which in my opinion is a technology to keep a close eye on. It is the official technology for Python which is considered to be one of the most developer friendly language because of their philosophy, and this has extended to the way they would create documentation. Another thing is that it is open-source and does not require a very complicated setup.

6.4 Conclusion

UX designer has increased in the UK, the US... it's becoming interesting for european countries. Yet, France industrials do not consider as seriously as these other countries. How we, ENAC student of the Master IHM can stand for more usability in the software development in France? Besides software development has starting to be outsourced for cheaper wages. Lived in romania... IT students should be concerned about this, as they will not be able to compete very long. I see two possibilities to maintain (interest), being an expert in a particular technology or starting to This is the kind of things I think would be beneficial for students to hear from our teachers.

Justifying is key to ux, and reporting is key for justification. My placement has lacked of reporting as it was difficult to understand what needed to be retracable and what not. Started with a board journal, but it's actually killing the information. Better is to focus on main steps like brainstorming, informal evaluation,

This report may take some strong position that better experts than me could easily critcise, and I would be happy to see them. I have just started to grasp to idea of a good UX design and this report can be seen as an effort to summarize my understanding.

This report has also been emphasizing the development side of the internship on purpose. UX designers are the interpret between users and developers. They should have a global understanding of computing as well as human behaviors. From my point of view, a good UX designer should be able to easily switch between platforms and limit his preferences, he should have also invested enough time to understand the tricks and ways of upcoming development process and that requires to deal with less user friendly tools. Yet, it's necessary to take this path. I am convinced that quality code and efforts made towards best practices lead to better design in the end by time saving, easy iteration and codeveloper friendly.

The work becomes research once the last sentence of the report written. Like, problem we allow people using results, but how about the device itself? Research can also consider the fact of making your project redoable.

Bibliography