DM

Predicting and Breaking Sedentary Behaviour at Work

Maria José Rodrigues Ferreira

A Nossa Universidade

Colégio dos Jesuitas
Rua dos Ferreiros - 9000-082, Funchal

Tel: +351 291 209400
Fax: +351 291 209410
Email: gabinetedareitoria@uma.pt

UNIVERSIDADE da MADEIRA

Predicting and Breaking Sedentary Behaviour at Work

MASTER DISSERTATION

Maria José Rodrigues Ferreira
MASTER IN INFORMATICS ENGINEERING

UNIVERSIDADE da MADEIRA
A Nossa Universidade
www.uma.pt

October | 2013

DM

UMa | Nome do Projecto/Relatório/Dissertação de Mestrado e/ou Tese de Doutoramento | Nome do Autor

DIMENSÕES: 45 X 29,7 cm

PAPEL: COUCHÊ MATE 350 GRAMAS

IMPRESSÃO: 4 CORES (CMYK)

ACABAMENTO: LAMINAÇÃO MATE

NOTA*
Caso a lombada tenha um tamanho inferior a 2 cm de largura, o logótipo institucional da UMa terá de rodar 90º , para que não perca a sua legibilidade|identidade.

Caso a lombada tenha menos de 1,5 cm até 0,7 cm de largura o laoyut da mesma passa a ser aquele que consta no lado direito da folha.

UNIVERSIDADE da MADEIRA

UNIVERSIDADE da MADEIRA

Predicting and Breaking Sedentary Behaviour at Work

MASTER DISSERTATION

Maria José Rodrigues Ferreira
MASTER IN INFORMATICS ENGINEERING

SUPERVISOR
Evangelos Karapanos

Acknowledgments

I would like to thank to my advisor Evangelos Karapanos for all the help and the guidance given. Thank you for the advice on the best options to take the multiple suggestions during the course of this thesis. For the explanations on how to use statistical software and how to interpret the data collected. For providing all the materials need to build the final prototypes and for helping me with my written and spoken English which helped me better myself.

I would also like to thank the PhD student Rúben Gouveia, for helping during the whole year, especially with the surveys, user interviews, and analysis of the data obtained during one study.

A special thanks to Délia Canha Gouveia Reis PhD in Probability and Statistics, from the Centre of Exact Sciences and Engineering of University of Madeira, for explaining and helping with the analysis of the data obtained from the first study.

I would like to thank all my friends, colleagues and testers for the availability shown and for the motivation help given.

I'm grateful to my family for all the support and encouragement given, especially during the most challenging moments.

Resumo

O objetivo desta tese foi planear e desenvolver um sistema que pude a) deduzir a necessidade dos indivíduos em fazer uma pausa a partir do seu comportamento sedentário no trabalho, b) persuadir os indivíduos para fazer uma pausa através da utilização de diferentes técnicas de psicologia persuasiva. Foram postuladas três variáveis, nomeadamente, postura do indivíduo, nível de stresse e envolvimento com a atividade mediada pelo computador. Foi desenvolvido um programa e um estudo que poderia inferir essas variáveis usando uma câmara web e o registo de teclas pressionadas e cliques do rato. Descobriu-se que o programa conseguia prever a postura através da distância do ecrã e o stresse através do movimento detetado. Foi então criada uma fórmula geral que prevê a necessidade de um indivíduo fazer uma pausa, usando só os indicadores postura e stresse.

Uma vez definido o principal objetivo, construiu-se uma aplicação e um estudo que usa três formas de comunicar uma recomendação para pausa ao utilizador: *implícito*, "*mesmo a tempo"* e através do *ambiente*. O *feedback* implícito é feito no computador dos utilizadores através de mudanças subtis do fundo do ambiente de trabalho que fornecem pistas visuais. O *feedback* "mesmo a tempo" é composto por alertas no canto inferior direito do ecrã dos utilizadores. A esta implementação foi adicionada uma técnica de interação intuitiva na parte de trás do ecrã, onde os utilizadores podiam suspender uma notificação usando gestos simples. O *feedback* através do ambiente foi feito através da utilização de uma escultura *origami* posta na mesa do utilizador. A escultura reflete continuamente a postura do utilizador e executa movimentos rítmicos quando é necessário recomendar pausas. Um estudo demonstrou o sucesso global do sistema, com 69% das recomendações para pausas recebidas pelos utilizadores sendo aceites. O estudo revelou ainda os pontos fortes e fracos dos três mecanismos persuasivos.

Palavras-chave

Comportamento Sedentário, Postura, Stresse, Envolvimento, Necessidade de uma pausa, Intervalo.

Abstract

This thesis aimed at designing and developing a system that can a) infer individuals' need for a break from sedentary behaviour in the workplace, and b) persuade them to take a break through the use of different techniques from persuasive psychology. We postulated three variables, namely, individuals' posture, stress levels and involvement in their computer mediated activity. We developed and field-studied a system that could infer these using a web camera and a key presses and mouse clicks log. We found that the system could predict posture from viewing depth and stress from the movement detected. We then created a general formula that predicts individuals' need for a break using only the posture and stress predictors.

Once the first objective was set, we built and field-studied a system that used three ways to communicate a recommendation for a break to a user: *implicit*, *just-in time* and *ambient feedback*. The *implicit* feedback was operationalized through changes in the users' computer wallpaper that provided subtle visual cues. The *just-in time* feedback employed prompting at the bottom right side of the user's screen. In addition, we implemented an intuitive behind-screen interaction technique where people can snooze a notification using simple gestures. The *ambient* feedback mechanism employed an origami sculpture sitting on the user's desk. This prototype was continuously reflecting the user's posture and performed rhythmic movements when to recommend breaks. A field study demonstrated the overall success of the system, with 69% of the break recommendations received by users were accepted. The study further revealed the strengths and weaknesses of the three persuasive mechanisms.

Keywords

Sedentary Behaviour, Posture, Stress, Engagement, Need for a break, Break.

List of Figures

List of Tables

List of Equations

List of Acronyms

BIFMA – Business and Institutional Furniture Manufacturers Association

CHIL – Computer in the Human Interaction Loop

ESM – Experience-Sampling Methodology

HCI – Human Computer Interaction

HSDF – Height of the Square that Detect a Face

IID – Interface and Interaction Design

JavaCV – Java Computer Vision

JMF – Java Media Framework

LED – Light-Emitting Diode

NIOSH – National Institute for Occupational Safety and Health

OB – Observed Breaks

OpenCV – Open Source Computer Vision

OSHA – Occupational Safety and Health Administration

RB – Recommended Breaks

SI Mean HSDF – Scale of the Interval that belong the Mean HSDF

SI Median HSDF – Scale of the Interval that belong the Median HSDF

SYCMY – Scale Y coordinate that identifies the movement

TCB – Total of Calculated Breaks

USB – Universal Serial Bus

UTP – Unshielded Twisted Pair

WinXP – Windows XP (eXPerience)

XCIM – X coordinate that identifies the movement

YCIM – Y coordinate that identifies the movement

List of Appendices

Index

1. Introduction

Sedentary behaviour exists since ever, but it has increased a lot due to the intrusion of computers in the workplace [1]. About 20 years ago, computers were not in fashion, because they were too expensive and too big. With the technological enhancements computers and laptops became cheaper and smaller and nowadays most people have at least one. Even companies had adopted computers as a mandatory equipment to have in the office. Working with computers has become common, and the levels of sedentary behaviour are increasing because of that, as people stay seated for too long.

Every time a problem arises, there is an attempt to find a technological solution to it; some examples that try to discourage sedentary behaviour are: Breakaway, Fish ‘n’ Steps and FitBit+. Breakaway it’s an ambient display (see Figure 1.1), with a design inspired in body language, common in theatre and animation arts, to express emotions. This application collects information from sensors placed in the user chair about the time the user spend sat. The more sedentary the user is more slumps the sculpture will be and the more breaks he does more upright it will be [2].

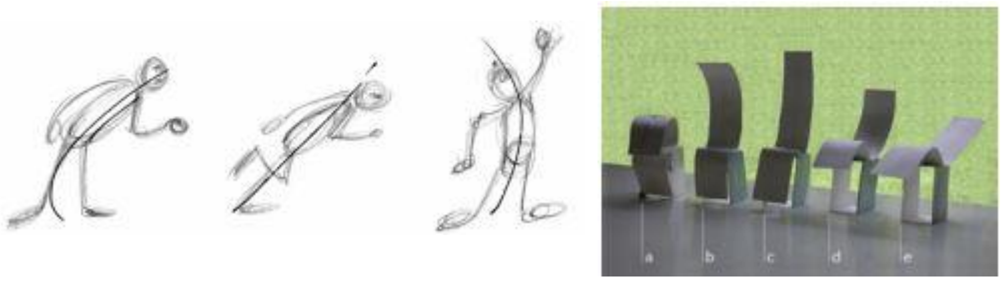

Figure 1.1 Breakaway - On the left examples of emotions created with Line of Action, on the right the sculpture design with Line of Action incorporated, adapted from [2]

Fish ’n’ Steps is a social interactive game application (see Figure 1.2) that provides incentives to people to increase physical activity through the growth of a virtual fish. The growth of the fishes, is related to the number of steps the user walks, measured with a pedometer, so the more the user walks, the more the fish grows [3].

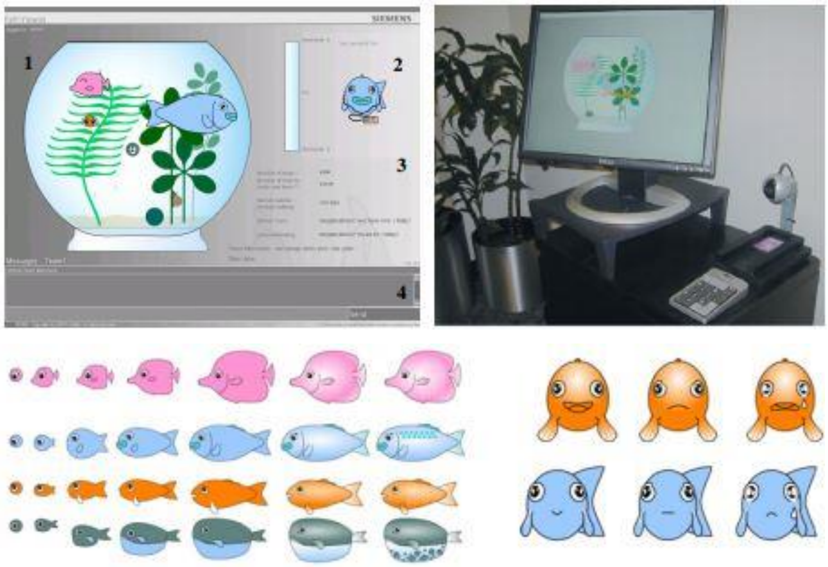

Figure 1.2 On top left, the Fish 'n' Steps user application, on top right the public kiosk and pedometer platform, and on bottom the growth levels and facial expressions of the fishes, adapted from [3]

FitBit is a device (see Figure 1.3) that tracks the steps, distance, calories burned and sleep through leading-edge accelerometer. This device can be synchronized to several equipment’s, such as computers, android phones and others. In these devices the user can see his progress with graphs, charts and reward badges [4].

Figure 1.3 FitBit devices, adapted from [4]

Since the sedentary behaviour had become a risk factor to human wellbeing researchers started investigating which are the main problems and the possible solutions to them. However, solutions found are too intrusive and have some limitations; they cannot analyse the activities users have in their daily routine (such as work tasks) or analyse for how long and what is the user doing on the computer. These limitations served as inspiration to develop an application for the Social Web course that would be able to overtake these issues and help individuals keep a healthy life. As it is difficult to modify people's behaviour when they already have very embedded habits, it is important to inspire them to abandonee these non-healthy habits. As a result, a micro-meetup solution called GetWitIt was developed with the purpose to inspire users through their friends/acquaintances network.

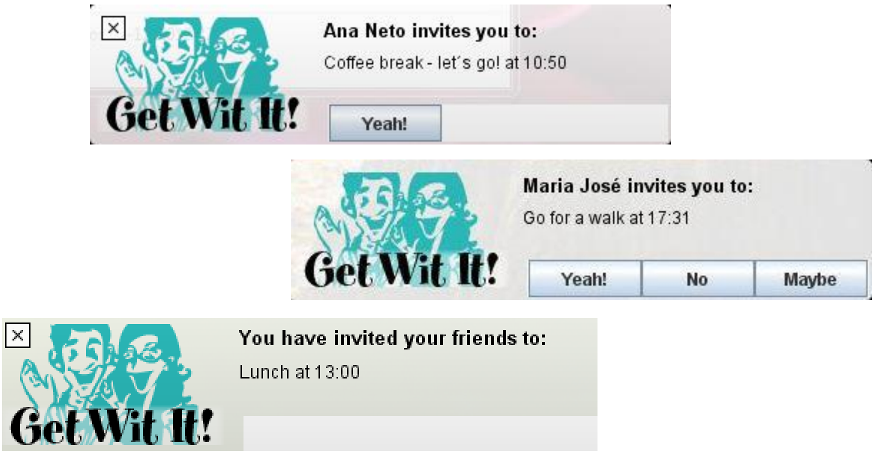

Figure 1.4 GetWitIt popups, on top 1-button version, in the middle the 3-button version and on bottom the creator notification

The GetWitIt program (see Appendix I - GetWitIt article, Appendix II - Entity–Relationship Model for GetWitIt application, Appendix III - GetWitIt Online Questionnaire) was developed as part of the Social Web course and had the purpose of encouraging people to do more breaks during the day through the help of other people that used the application; they could be friends, acquaintances or complete strangers. The strength of the application was in the notification system, one of the novelties was introduced in the buttons for acceptance/refusal choices in the notifications popups (see Figure 1.4). Some users preferred to be alerted ahead to be prepared to go to a break and others had no problem receiving always the same message as long as they knew the sender.

GetWitIt used a micro-meetup strategy for prompting activities that were specific, short-term and actionable [5]. We expected that users would choose activities that they are confident they can attain. Users receive cues-to-action through reminder messaging and the system design allowed them to experiment, seeing how they could integrate into their daily life.

The results collected with this application show that the main idea works, but people tend to be unpredictable. Letting the program be dependent only on the users input revealed to be the wrong approach to encourage people to be less sedentary. This approach does not have into account if the user needs a break and if the moment is the most appropriate to receive a recommendation.

The GetWitIt app was a good starting point to figure out what motivates people to be less sedentary and what could be the better direction to follow to build an application to help break sedentary behaviour.

1.1. Thesis Structure

In chapter 2 we motivate the importance of predicting user's context and specifically their need for a break, before recommending a break of sedentary behaviour. To predict the better way of recommend a break, it was essential to understand what types of systems already exist and what researches have conducted in this area. The methodologies present in the literature review served as support, either to replicate or to set aside, what is the best way to break sedentary behaviour without disrupting the user workflow.

Chapter 3 reports a study that aimed at predicting users' posture and stress levels using features derived from the web camera as well as the continuous monitoring of users involvement based on mouse and keyboard activity. The goal was to infer users posture from on the viewing depth from a user to a screen; stress from the movement performed by the user; and the involvement from the actions when using the mouse and the keyboard. Based on the three previous characteristics the intent was to analyse if the need of a break is likely. The results collected from the study were positive for three of the four original hypotheses and it made creating general formulas to predict posture, stress and the need for a break possible.

Chapter 4 presents a system that monitors users' levels of posture, stress and involvement at the computer tasks and attempts to motivate users to break sedentary behaviour through the use of three mechanisms: implicit, just-in-time and ambient feedback. The implicit feedback uses wallpaper changes to alert the user of his current state. To represent the user states of sedentary behaviour three levels of stress were implemented (low, medium and high). This was based on the time the user was sat at the computer. The just-in-time recommendations were characterized by the existence of two extra pieces – the screen and the sculpture prototypes – to give feedback to the user concerning a break alert (both prototypes) and his current posture (just sculpture prototype). The ambient feedback used the sculpture prototype and had the purpose to provide influence from the sculpture movements, not just in the user but also the surrounding people.

Chapter 5 presents a study with 4 users over 10 days of use of the application described above. This application implements formulas that predict posture, stress and the need for a break collected from the study of chapter 3. The goal of this study is to understand the impact that the program had on users when they were presented with different mechanisms that recommended a break.

1.2. Contributions

With this work we developed and field-studied two systems. In the first study we postulated three variables, namely, individuals' posture, stress levels and involvement in their computer mediated activity. The goal was to achieve the users need for a break by the inference of the variables using a web camera and the log of mouse and keyboard activity.

In the second study we implemented three ways to communicate a recommendation for a break to a user: implicit, just-in time and ambient feedback. The goal of this system was to analyse if the feedback provided would have some influence on the user's acceptance to do a break from sedentary behaviour.

2. Predicting breaks

Knowledge workers spend between 4.5 to 9 hours seated on a daily basis [6]. This *sedentary behaviour* (i.e., spending considerable time seated and along with low levels of energy [7]) is the cause of considerable health problems, such as eye strain, neck and shoulder pain [8], metabolic disruptions, obesity and premature mortality [7].

In order to encourage individuals to break sedentary behaviour during working hours, we aim to build a computer system that estimates appropriate times when individuals might want or need a break. Because prompting at inappropriate time might disturb the user concentration and workflow. User's workflow could be disturbed especially if the notifications were: about unimportant information; too frequent; a flood of notifications and have annoying persistent sound [9].

To predict breaks, it is important to get answers to the questions: How do people know if they want or need to take a break? How can these systems be created? How can be stress and posture predicted? With this questions it will be easy to understand what are the people needs and how can a system help them in order to help break their sedentary lifestyle.

2.1. Availability and/or Interruptibility

Availability and interruptibility are two important concepts to have in mind when thinking if someone needs a break. But what does availability mean? To be available means that someone it’s receptive to receive any kind of disruption in his/her work without perturbing the current task. On the contrary interruptibility means to break the continuity of an action/event. These two concepts are not the same, because one reflects the willing/acceptance to be disturbed and the other it’s related with case with which an action/event can be stopped.

To see when a person need a break it’s important to understand if she is available or not and/or if it can be interrupted or not. The level of availability and interruptibility it’s dependent on the current task of the person. For example if a person is writing an article and its very focus her availability to speak and/or stop the current reasoning probably will be very low because this might break her concentration.

To create systems that can predict if someone it’s available and it can be interrupted can be very difficult, because unfortunately, computer cannot perceive information like people can - from emotions and behaviour observation. Computer and communications systems are almost entirely unconscious to the human context and cannot judge whether “now is a bad time.” [10]

Lilsys and MyConnector were developed for office environments in order to help predict user’s availability. These systems were developed to track information from several sensors and some communication devices. Sensors can be used in different ways and can track a lot of different information, Hudson, et al. [10] study tries to understand which sensors are the most accurate to predict human interruptibility.

Lilsys is a prototype system that uses ambient sensors in predicting the user’s unavailability. In this paper the authors argue that a system that is working – left online – does not want to say “someone is here” or nearby. People tend to leave their computer unlocked when they are shortly absent, or when they work in the surrounding environment, so having the computer on does not mean someone is using it or it’s in the proximity. In Lilsys they use a set of sensors, such as, motion, sound, phone, door, and computer combined with symbols to show to other users the availability status of the user (see Figure 2.1).

Figure 2.1 Lilsys prototype, adapted from [11]

In this system they try to examine notions of presence and availability and problems related to determining availability in telecommunications. In the end of this analysis they concluded that machine interpretation of ambient sensors was detecting cases of lower availability; they detected issues in: portrayal of usability assessment; hardware interface elements; user image control and asymmetric capabilities. One of their findings contrasted their expectations, despite what unavailability assessments show the interruptions were no less frequent.

Although Lilsys system was not 100% accurate in the predictions made, it was more assured to detect non-presence, when using the combination of motion, sound, phone and mouse plus keyboard activity. This information was compared with the Awareness system that only used keyboard and mouse monitoring to detect presence. [11]

MyConnector it's a context-aware proactive service used to facilitate the communication based on availability of the two interested parts. This system display availability of a person based on her calendar entries, affording a number of actions such as calling, emailing and others. This program uses the framework of the project CHIL – Computer in the Human Interaction Loop that had the goal to create services that can sense human activities, interactions and intentions. My Connector it's available for standard phones, smart phones, desktop application (WinXP – Windows XP) and web clients (see Figure 2.2). They use a Bayesian learning approach to learn availability of a person. It was collected information of the personal calendar with the Microsoft Exchange server; the desktop application collects mouse, keyboard and Skype activity. In the desktop application they collect too information about general availability for different communication media (for example if the user has physical access to cell phone, etc).

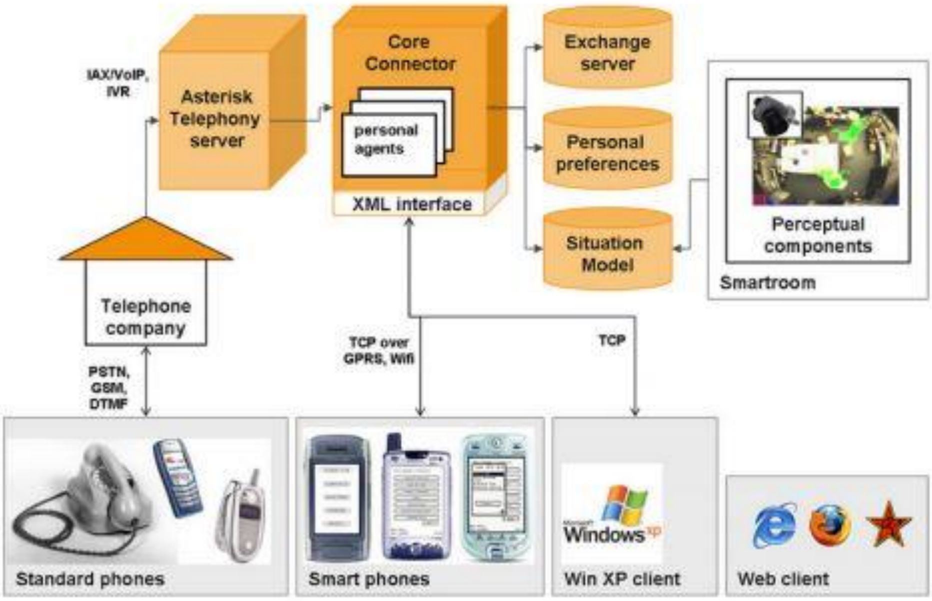

Figure 2.2 MyConnector architecture, adapted from [12]

Results on this pilot study, show that PC activity – mouse plus keyboard activity, frequency of windows switching and active programs – presented lower results than the expected to predict availability. But with the MyConnector active, it revealed to be the best indicator for detect availability. Information about person's activity like urgency and importance of current activity it was extremely valuable. With this information they achieved two different approaches. In the first approach they assume availability is too complex and difficult to be automatically learned and sense. So the senders would decide whether it's the best appropriate time to contact the receiver. In the second approach they assume all intelligence is in the system so he decides in behalf of the receiver [12].

Predicting interruptibility was attempted by Hudson, et al. [10] in a study that explores which sensors are the best interpreters to use in this situation. It was used human coding of audio and video recording as ground truth and filter for non-relevant information (for example, after encoding the number of people present in the room, there were no visible sequences of images without occupants). In this study they used different learning algorithms to construct different forms of predictive models to achieve and answer to which sensor(s) it's better to expect interruptibility. They try to get answer to five questions: "Can a practical sensor-driven model reliably predict human interruptibility? How can such a model be constructed? How accurate can we make such model? Which sensors are most useful for such model? What are the simplest sensors that will produce an accurate prediction?" For this study they used four subjects and all received a PC with large disk, an A/V capture and comprehension card connected to a small camera and microphone.

It were created 23 simulated sensors based on information logged from the coding done in the recorded data and grouped in three categories, occupant related, guest related and environment. Some examples of the simulated sensors are: occupant presence, interaction with desk and other nearby material, sitting, writing, number of guest present, door open or closed, etc. They used several learning algorithms such as Decision Tree, support vector machine, naïve Bayesian predictors and AdaBoost with decision stumps, based on the binary decision problem with two states, 'interruptible' and 'non-interruptible' using the simulated sensors with more information.

The results collected from the above models were similar and didn't produce statistical significance. Since different unrelated approaches produce similar results they hope that robust models with 75-80% accuracy range can be driven from real sensors. The best modelling technique was provided by the Decision Tree model with an accuracy of 78%. Outcomes from and *information gain metric* based on the features collected – simulated sensors –found that speech and movement detection are the top two sensors with higher predictive power. But this information not shows if these features indicate interruptibility or non-interruptibility [10].

The existing systems and researches have some problems because they might be too intrusive for the user and/or too costly. Having systems that decide in behalf of the user might become too aggressive and make them feel presser to do something or completely ignore the system. The cost of the system its influence by the quantity of extra equipment that is need to make it work more accurately, so the more equipment its need to detect availability and/or interruptibility the more costly it's the system, making it inaccessible for all user.

2.2. Stress and Involvement

Stress and work-task involvement are two attributes that go hand by hand. The European Agency for Safety and Health at Work define stress as an imbalance that people suffer between the demands made of them and the resources they have available to cope with those demands [13]. In a working place stress sometimes can be “visible” by the person actions and behaviour, through what is called *fidgeting*, the act of moving about restlessly [14]. When individuals feel more pressured, they are unable to control their stress and the situation, and in most of the times, they do not have conscience of their actions. Fidgeting examples can be seen in Figure 2.3.

Figure 2.3 Examples of fidgeting, adapted from [15]

Fidgeting is one of the easiest ways to understand if people are under stress. In a computer workstation, fidgeting often extends to users’ interactions with the computer. Computer “fidgeting” is most usually observed through an excessive amount of mouse clicks and, sometimes, also keyboard presses.

An increase of mouse and keyboard activity does not necessarily suggest increased levels of stress; it may also relate to users’ heightened engagement or involvement in their computer task. Involvement is broadly understood by many as the act of giving a lot of time and attention to something you care about [16]. Working at the computer implies performing different tasks; some related with leisure, others with the job and normally people feel more involved and focused when they like what they are doing. For example if someone it’s playing his favourite online game with friends, his levels of involvement would probably be higher. The harder the task and the less the intrinsic motivation for the task, the lower the expect involvement on the users’ side.

It’s hard to identify emotions like stress, frustration, nervousness, tiredness and other with systems, but studies like the ones from Navalpakkam, et al. [17] and Epp, et al. [18] are trying to obtain user states based on specific tasks and actions in the computer.

Mouse and Eye tracking were combined to detect in a web page the user’s frustration and reading struggles. Navalpakkam, et al. [17], described how the use of those two data sources in the evaluation of content layouts more specifically the position of an ad can contribute for a better reading experience. They try to identify mouse markers that can predict user frustration and reading struggles at reasonably high accuracy. They performed two different experiments, each with 90 and 20 participants respectively. [17]

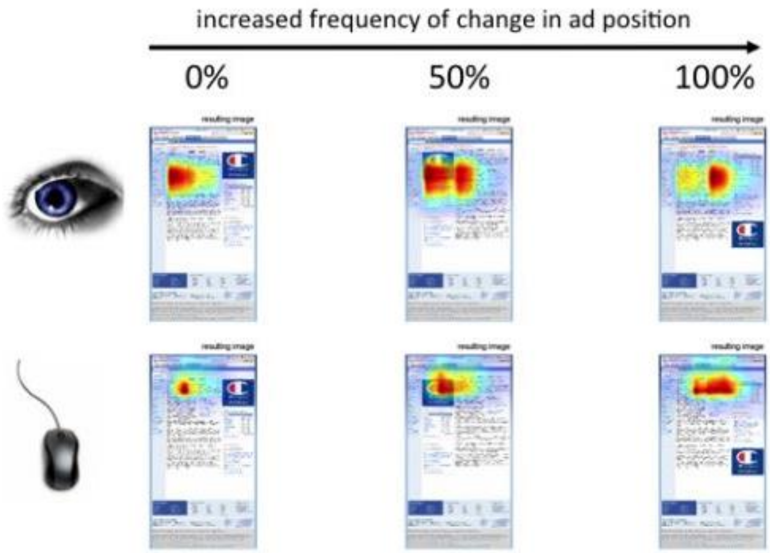

Figure 2.4 Changing positions of the ads, heat maps on top for eye tracking and in bottom for the mouse tracking, adapted from [17]

The first experiment had three treatments, each had 20 participants and all participants saw six articles. In the first treatment the ad was always on the top right page; in the second the ad was at the top right in 50% of the cases, the other 50% they were random at one of the five remaining positions; in the third treatment the ad could appear at any position of the page (see Figure 2.4). In the end of this experience user give their feedback and fill a survey. It was used standard algorithms to parse eye track data to obtain eye fixation. [17]

Results show that changing layouts with frequency has impact in the way people engage with content and the ad. When changing the ad position the eye detects it 70% sooner and the mouse 60% sooner comparing to when position is fixed on the page. [17]

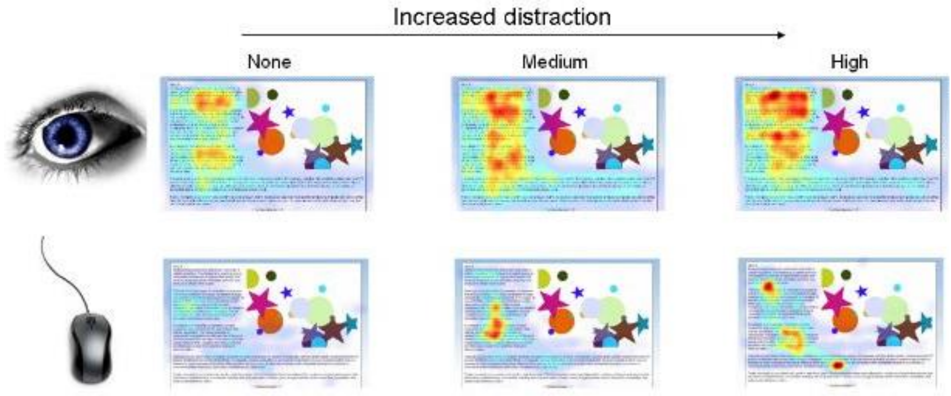

Figure 2.5 Impact of distractions in the ads, heat maps on top for eye tracking and in bottom for the mouse tracking, adapted from [17]

The second experiment (see Figure 2.5) had three factors: no distraction, medium distraction and high distraction created by presenting no graphics, static graphics and animated graphics. In this experiment participants saw 3 essays and answer to five questions for each essay. The results for this experiment show that when distraction levels increase, participants have more difficulties reading the text. The eyes gaze and mouse activity it's higher in this cases. The report experience measured in a five points scale that mouse can predict pleasantness experiences 15% better than chance. The eye gaze and mouse patterns contain rich information about the user state. These findings suggest that mouse tracking could offer a scalable way to infer user attention and experience on the web [17].

Epp, et al., [18] study tries to answer problems like: wrong interruptions, inappropriate feedback and increase of frustration by making advances in: set of mechanisms for collection and modelling user context, set of techniques for adapting user interfaces and system behaviour based on contextual information. This type of system that detect and respond to an emotional state could improve user performance and satisfaction. [18]

In the study they executed an experience-sampling were users entered an additional sample of fixed text from Alice's Adventures in Wonderland and also categorized their data with their level of agreement scaled with 15 emotional sates. [18]

Areas such as affective computing are interested in emotional states from users; to define emotions it has been performed two approaches, the categorical and the dimensional. The categorical approach it's related to give appropriate labels to different states through language. The dimensional approach uses two orthogonal axes, the arousal and the valence; arousal it's related to the emergency of feelings and the valance describe the pleasure or displeasure of the feelings. To measure the affective state of user's using interactive technology HCI – Human Computer Interaction researches had used physiological sensors but this approach requires the use of specialize equipment and it's too obtrusive. [18]

To study unique timing patterns in an individual typing it was used keystroke dynamics, in this type of studies normally are extracted some features in keystroke timing such as: duration of a key press and elapsed time between key presses. This kind of research had been made previously in authentication systems and they show strong evidences in the interference of authentication because of strong emotional states. [18]

It was used an ESM – Experience-Sampling Methodology to obtain keystroke as well labels to each data point through responses of an emotional state questionnaire. The purpose was to receive emotional information's in real-world time and explore more emotional states. To help users accomplish this they needed to use a device that alert them to complete the self-report. It was used a C# program to collect keystrokes measurements regardless of the focus application. The users had the hypothesis to choose or not to exclude the prompted data from the sample of the last 10 minutes. To collect current emotional state of user (example I am frustrated, I feel confident, I feel tired, etc) it was use the questionnaire with answers measured through a 5 point Likert scale. [18]

With the data we get three categories of information related to: keystroke/content features, emotional state classes and additional data points. The data related to keystroke involved was: events of key press and release, unique codes for each key and time stamp when a key event occurred. In the keystroke features they had measurements such as: keystroke duration feature (dwell) - for single key features and graphs; keystroke latency feature (flight) – always involved two keys; other keystroke features – features that combine latency and duration (example key down to down); keystroke feature overlap – some features overlap slightly. There were some specific features for the free keystroke, such as: number of characters, numbers, punctuation marks, uppercase characters and the number and percentage of "special characters". For both free keystroke and fixed text it was collected the number of mistakes (backspace + delete key). In the classification of the emotional states they group up the results for only 3 classes agree, neutral and disagree; the "strongly agree" and "strongly disagree" were group in the "agree" and "disagree" respectively. [18]

For identifying emotions it was used the decision tree algorithms approach, and to choose prominent features for the emotional states models they use the correlation-based feature. It was used the under-sampling method that randomly takes out instances from majority classes to make them stay equal to the ones with minor instances to create a uniform distribution. To emotional states that suffer this method they performed 10 times the classification process and collected the mean classification accuracy and variance. [18]

In this study it was made separated models for fixed text and free text, but it's only presented models about fixed text. Using their evaluation categories they consider that models that accomplished their top three are a top emotional state. There are some emotional states ("tired" and "relaxation") that had lower variance in the classification rate and kappa statistic, this could mean that this model results are reliable indicators of the predictive power of keystroke dynamics for emotional state decision. [18]

The results presented in the article show that keystroke dynamics can perfectly classify from seven emotional states (confidence, hesitance, nervousness, relaxation, sadness and tired) at least two states. These results also found two new potential states for this type of classification, anger and excitement. The current software doesn't "know" when the user might be in a low emotional state to prompt them with the experience sampling questionnaire so the prompts are made at random times. [18]

Extracting emotional aspects of situations from a system would be very good for making right decisions about how to interact with the user or adapt the system response. Knower days this kind of approaches have some restrictions because they can be too invasive as well require expensive equipment [18].

2.3. Posture

Posture it's a position of the body or of body parts [19]. Posture must be taken into account in almost all movements and actions, because severe health problems might show up by making "bad moves". The most common problems associated with posture are related with the head and the shoulders (see Figure 2.6).

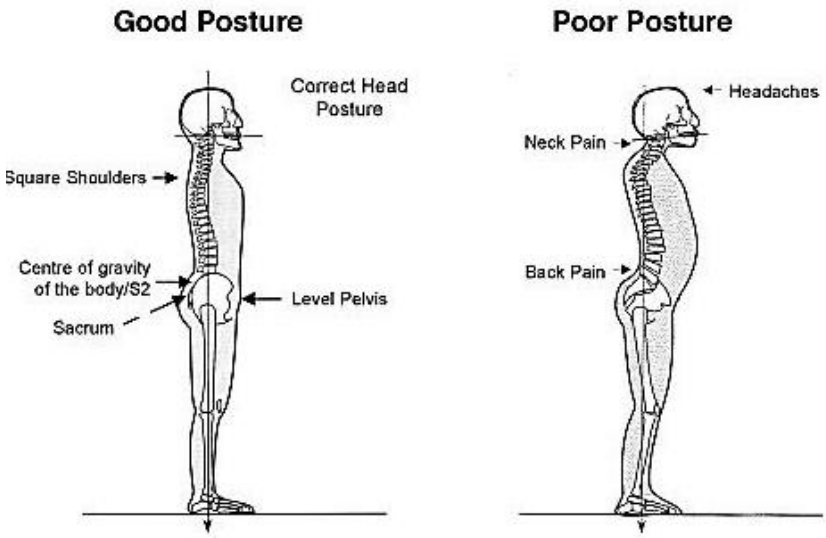

Figure 2.6 Examples of good and bad postures, adapted from [20]

Why posture it's something important when working at the computer? The importance of posture when people work in a computer it's related with the fact that bad postures might get a bad influence at least in user productivity.

There are several things that affect negatively the user posture when he works in the computer and they do not depend exclusively from him. Computer workstations furniture it's a good example of something that normally does not depend on user choice and might produce a bad influence when does not follow the standard requirements to prevent injuries in users.

Ergonomic books are a good source of information to better understand how to identify good and bad postures, what type of equipment can be used and the types of measures that can be taken in order to correct/improve the user health. In the guidebooks of Office Ergonomics – Guidelines for Preventing Musculoskeletal Injuries and Ergonomics and Design a Reference Guide it can be found some recommendations to identify: if you work in an appropriate workstation, if you are at a good distance from screen, if you are well seated and so one.

The guidebook from WorkSafeNB [21] has the purpose to help people identifying if their computer workstation is appropriate for them or not.

The first thing to do it's to know how to identify what a good posture is. A chair, the work surface, the accessories, the monitor and the input devices can help you, or hold back in maintaining good posture. An ideal posture should be defined as: head upright and over shoulders; eyes looking slightly downward without bending from the neck; back should be supported by the backrest of the chair; wrist in a neutral posture; elbows bent at 90º, forearms horizontal, shoulders should be relaxed, but not depressed; thighs horizontal with a 90º - 110º angle at the hip; feet fully supported and flat on the floor, if not possible supported by footrest (see Figure 2.7).

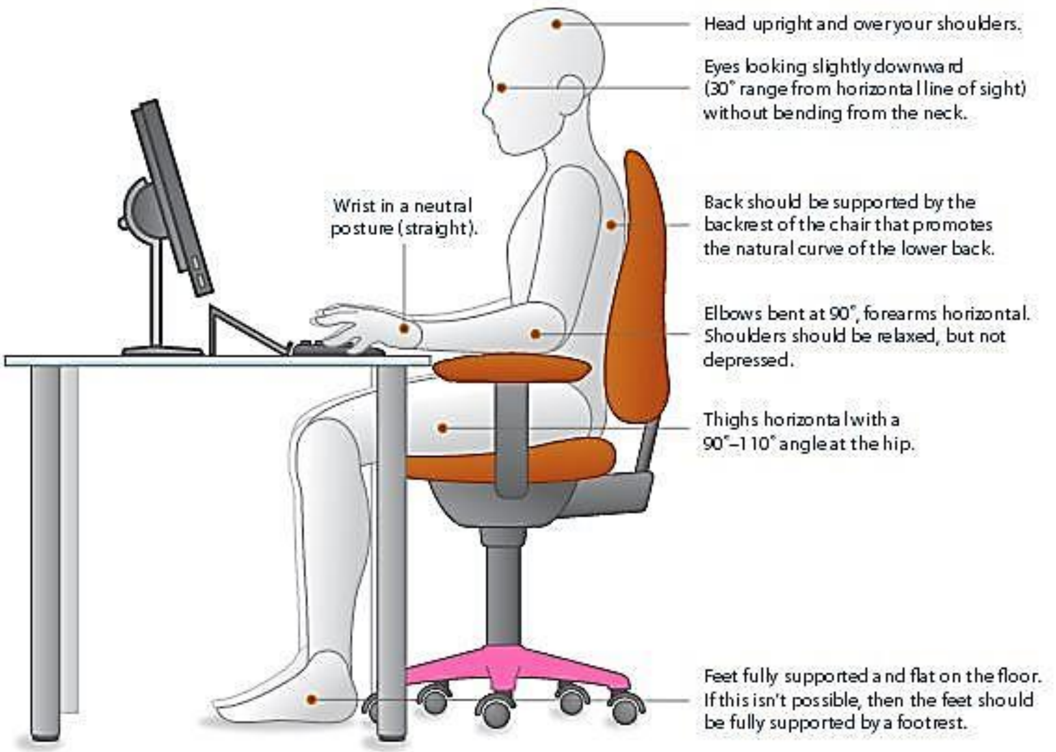

Figure 2.7 Good seated posture at the computer, adapted from [21]

When something is wrong in a workstation the first thing that people look is their chair. Some basic guidelines are: to provide lumbar support, height can be adjusted and etc. Once they adjust the chair, they can determine the appropriate height for their work surface. Materials used frequently should be located within easy reach (a good way to arrange work materials is in a semicircle shape as in Figure 2.8).

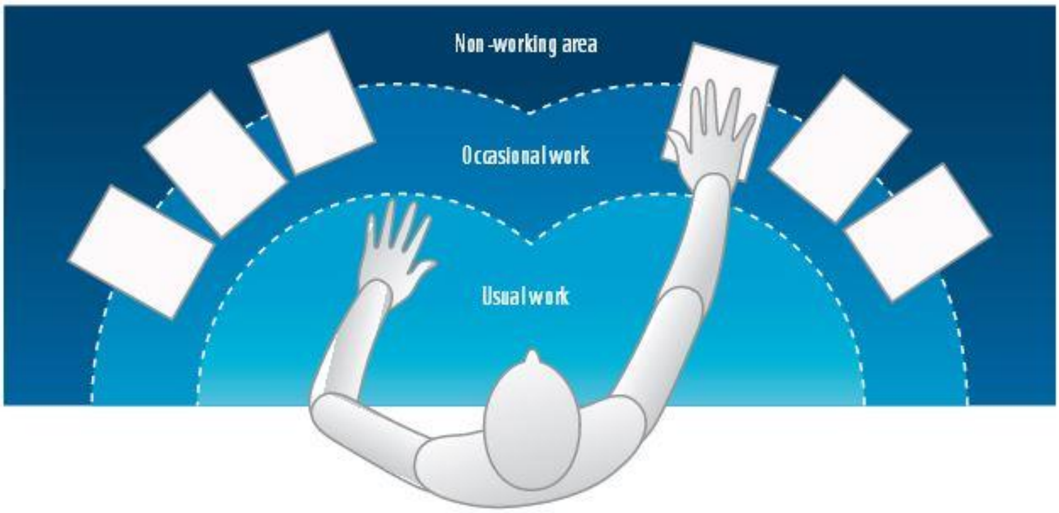

Figure 2.8 Layout of a working area, with colours representing areas of frequent work, adapted from [21]

Studies have found that monitor distance should be between 60 – 90 cm, if you straighten your arms in front of you and touch the monitor; probably you are too close of it.

A laptop computer when use often the same principles apply as with any computer workstation. If you have the chance use a docking station, these will help you turn your computer into a desktop computer.

Always as possible people should do stretches (every hour or throughout the day). Some examples of stretches: shake out hands, shrug your shoulders; raise one arm above your head switching arms and etc (see Figure 2.9). [21]

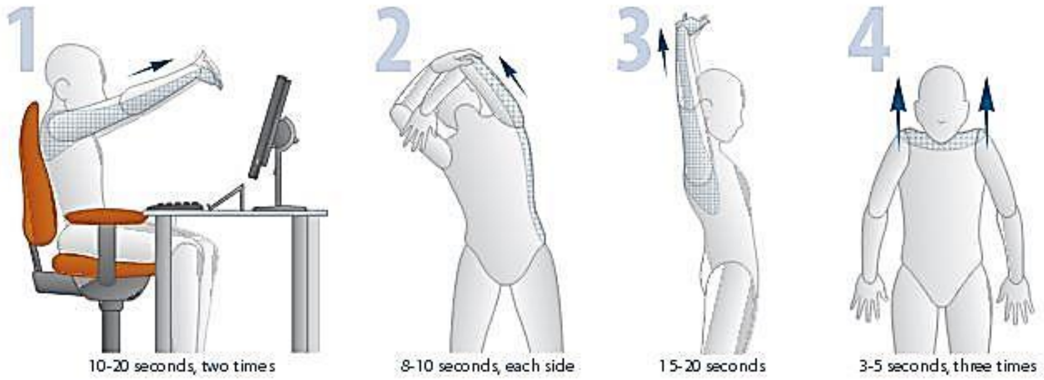

Figure 2.9 Examples of four stretches, adapted from [21]

In three sections of the guidebook of Openshaw,S. Allsteel;Taylor,E. Allsteel, [22] it's approached the common workplace posture, the common workplace motions and office furniture guidelines for fit and functions. In these sections it can found some common measurements for postures when standing, sitting, reaching and moving.

When a user stand, the desk height could be in a range of 71.1 – 109.2cm depending on the type of desk work (precision, light or heavy), and be aware that the computer monitor should be located at eye level or slightly below (see Figure 2.10).

Figure 2.10 Worksurface height depending of the work executed, adapted from [22]

Users that use to be in seated position should follow the common anthropometric measurements that are slightly different for female and male. In Figure 2.11 it's represented by **A** and **F** letters examples of two of these measurements: sitting height (female range 79.5 – 90.9cm; male range 85.3 – 97.3cm) and knee height (female range 50.3 – 58.9cm); male range 54.4 – 63.5cm).

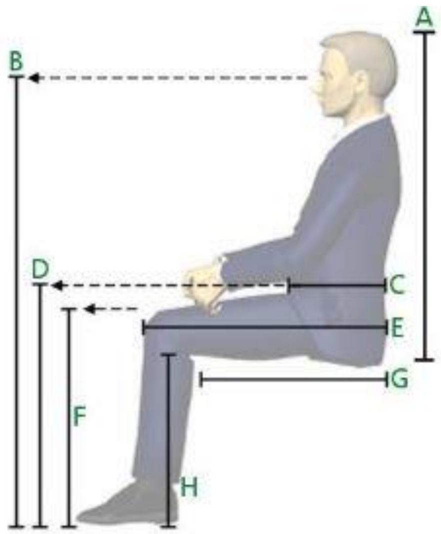

Figure 2.11 Common anthropometric measurements for the seated position, adapted from [22]

When users are in a good seated posture the should be comfortable and that don't put stress or strain on the users buttocks, back or arm muscles, and allow them to put the feet on the floor.

In a workstation people need to reach things nearby and they should allow the majority of the movements of the user's body joins within healthy zones (see Figure 2.12 examples of two join movements and Appendix IV - Zones of Motion joints) in order to minimize awkward or healthy positions.

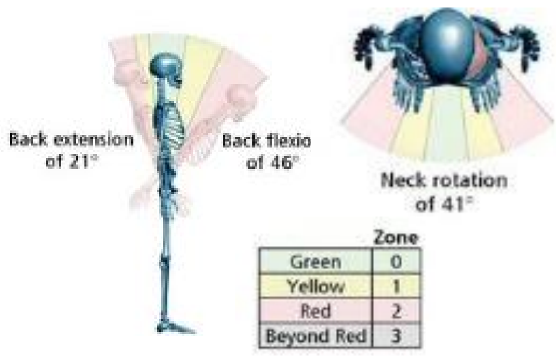

Figure 2.12 Zones identification for the back and the neck, adapted from [22]

Operators work can be support using adjustable desks when considering heavy work, the table can be at a height that allows good leverage for lifting and moving objects. For precision work, desk height can be higher than normal because the worker will need to focus closely on the workpiece.

The placement of some accessories such as keyboard tray and mouse pad holder can allow users to put them in comfortable positions where there is minimal strain to the back, arms and wrists. For example one measurement considers supporting surfaces for computer desks its viewing depth, which according to BIFMA – Business and Institutional Furniture Manufacturers Association guidelines should be greater than 39.9cm from the visual display terminal to eyes, in Figure 2.13 the **H** letter). [22]

Figure 2.13 Desk and worksurface placement, adapted from [22]

How can ergonomic concepts like posture and distance from screen be useful to track the user need for a break? Persons that work at the computer for large hours during the day do not stay at the same position the whole time. The stress, the tiredness, the eye fatigue and other factors have influence in the person posture during that period.

Chen, et al. [8] article shows how an application can be created using a collaborative framework for ergonomic feedback using only web cameras to track all the information into an office environment.

The agencies OSHA – Occupational Safety and Health Administration – and NIOSH – National Institute for Occupational Safety and Health – are increasing their promotion about the importance of proper ergonomics for improve office workers health and wellbeing. Chen, et al. [8] describe a collaborative framework that uses a computer's web camera and cameras in the workplace environment to collect workers current feedback about their ergonomic state. With the created program they collect measurements such as: worker average, work breaks periods, distance between worker and his monitor and measurement about worker head motion. The extra cameras spread around the office environment provide information about posture and social interaction (see Figure 2.14).

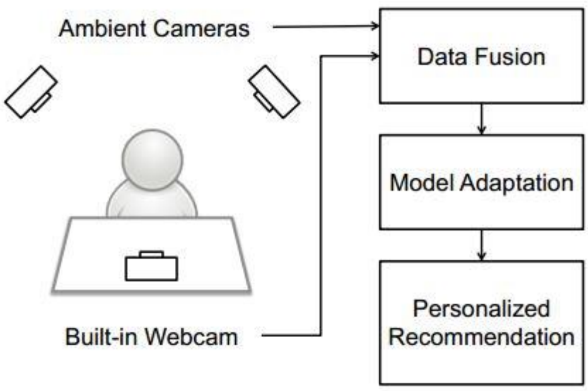

Figure 2.14 System overview, adapted from [8]

Nowadays ergonomic experts are developing strategies and references not just for workers but for employers too in order to reduce injuries in the workplace. Self-conscience about bad habits in workers it's important otherwise they will be reluctant in follow the guidelines provided to correct it.

The system created collects from face detection and head orientation information about presence, motion and attention.

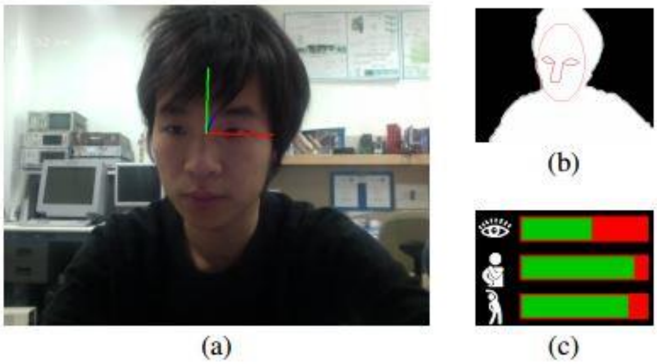

Figure 2.15 Information presented to the user, adapted from [8]

In Figure 2.15 there is an example of what is shown to users when they used the application, (a) it’s the video from the computer web camera, (b) the user’s silhouette and eye location, (c) graphical health bars – duration of use of the eyes, head mobility and presence in front of the computer.

Users can be alerted by the web camera feedback and by additional notifications. The web camera provides information through face detection that estimates posture and eye fatigue. The posture it’s predictable by the foreground silhouette and the eye fatigue by the blink rate calculation. The reminders shown to user can inform him about distance from screen, to rest the eyes and other pertinent information.

Nowadays there are applications that prompt reminders at predetermined time, but bad timing might interrupt the user work and affect negatively their efficiency and productivity. Systems must be intelligent in order to understand user’s behaviours to make reminders pertinent to them. The advances made in ubiquitous sensors and computing power had made possible gather complete personal information.

Eye tracking and gaze estimation have been used in several programs, and are fields that are well studied, because it’s becoming important to understand where people are looking at. One of the biggest problems of these systems it’s because they can be too costly and/or too intrusive for users. With the computer web camera it was tracked faces and gaze detection; the algorithm used to identify faces possesses a high range of detection in various poses, this because users do not stay too static along the development of their usual tasks. The face detection provides information to calculate the distance that users are from the screen (to understand computer viewing habits) and the measurements of the blinking detector (the head and the estimated eye location).

Foreground segmentation is an aspect of computer vision which is well developed, but even good algorithms that can predict motion base segmentation can fail if an object remains motionless for too long, because the object can be “merge” into the background. Estimation of human posture it being made a lot through the human silhouettes. In the developed program it's made an assumption of the approximate silhouette of the foreground object in order to identify clearly a person from the background.

The philosophy used in the system created was: less is more; in order to spare computational resources. The system extracted in real-time ergonomic information that can be interpreted by the user using the application or an expert in the area. [8]

2.4. Conclusion

In this chapter we wanted to get answers to three questions in order to understand what are the people needs and how can systems help them in order to help break sedentary lifestyle.

How do people know if they want or need to take a break? In order to understand if a person need to do a break it’s important to understand if she is available at the moment and if she can be interrupted.

How can be stress and posture predicted? The stress can be predicted through the movements that a person does and posture can be related with the viewing depth from the computer screen.

How can these systems be created? The creation of this type of systems can involve sensors and specific algorithms (such as motion detection) implementation.

3. Study I – Predicting posture, stress and engagement through the web camera on computer tasks

Nowadays people are even more dependent of the use of computers that sometimes they lose track of time. Unfortunately it's not only time they lose track, occasionally they don't realise that they are sit in a wrong position for too long. There are some ergonomic guidebooks that explain how people can sit and work in a good posture when they are in front of the computer. What distance they should be from the computer, the layout that most frequent materials should have and other important factors. People that use to stay in bad postures for too long might be affected by several problems that can provoke injuries for the rest of their life, in the most severe cases.

It's not only bad postures that can jeopardize work efficiency, stress can be too a bad "ally". The rise of stress can be related with the current task of the user or external factors to that, and it's important to understand if a person is feeling stress or not. One of the most common "visual effects" of stress are fidgeting's. Fidgeting is associated with movement, so the more stress a person is feeling probably more movements she is going to do.

When a person works with the computer it might suffer of "computer fidgeting", sometimes people might feel more involved in their daily basis task and others not really; it depends on the task itself. Normally people when feel more involved tend to touch/hit in what they have nearby, normally mouse and keyboard are the closest object, so click more and/or press more the mouse and keyboard respectively might be one of the fidgeting "effects".

With knowledge about posture, stress and engagement effects when people work in the computer for too long, it was important to understand if it was possible include these measurements into an application. To accomplish this we developed a Pilot Study that collected pictures and proceed to the identification of the motion detection. In this applications it was used the library OpenCV, the interface JavaCV and the Java Media Framework – JMF - to accomplish the above (see Appendix V - Installation steps for: OpenCV, JavaCV, Java Media Framework). With this application we performed an update and added new features to it because we had the purpose to get a proper understanding if it was possible to collect measurements that could represent posture, stress and engagement using only the web camera. The improvements made in the reuse code were in order to correct some bugs and make it work more efficiently. The new features added are related with face detection, distance detection and behaviour analysis.

It was chose university students to perform this study because they don't have a fix workload along the weeks and they spend a lot of their time studying, doing papers, practical works and others in the computer.

In this research it will be try to get answers to the following questions: Can posture be predicted from distance to screen? Can stress be predicted from motion detection? Can involvement be predicted through the mouse and keyboard activity? The need for a break can be predicted through posture, stress and involvement of the user? If the results achieved were positive for more than one of the hypothesis presented they will be apply in the application Break the Sedentary Behaviour through feedback (see section 4, page 51) to improve the non-sedentary behaviour of a person.

3.1. Pilot Study

The Pilot Study had the purpose to understand and see if combining the mouse, keyboard and web camera would help foresee if there is someone at the computer. In this research it was analyse the behaviour that individuals say they had with what they really do, and understand if their normal work depend or not on the use of the computer.

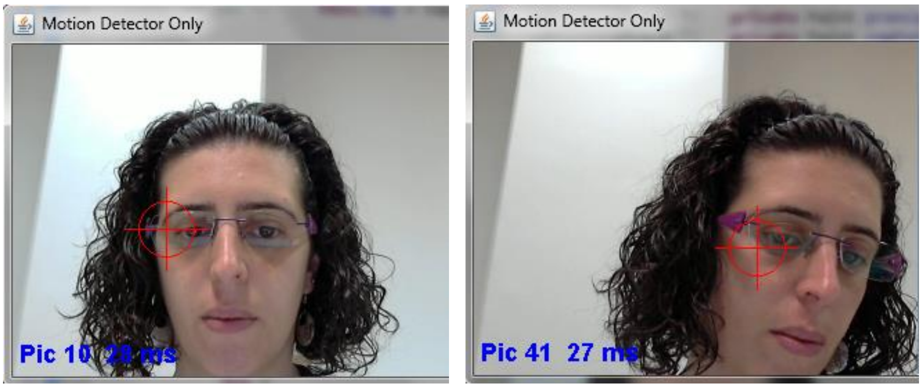

Figure 3.1 Detection of movement on pictures with seconds of difference

The system created for this pilot experience (see Figure 3.1) used motion detection to analyse if it was detected any movement while the program was running (see Appendix VI - Process to detect Motion), storing all the information into an online database[1] (see Appendix VII - Entity–Relationship Model for Pilot Study). The pictures taken by the motion detector were all stored in the user's computer to be verified the accuracy of the detection made. It was collected too mouse and keyboard activity in order to identify absence of the computer when combined with motion detection information.

[1] Online database allocated in www.apus.uma.pt server.

Figure 3.2 Pictures taken with the application when the user was on a break with half second of interval, the right picture contain red indications of movement compared with image on left

One of the biggest problems detected with this study were the background environment choose by users, with access to the outdoor. Every time that there was wind and the user was away from the computer (see Figure 3.2) or wasn't performing great changes in his behaviour (see Figure 3.3) the program captured the movement of the swaying trees.

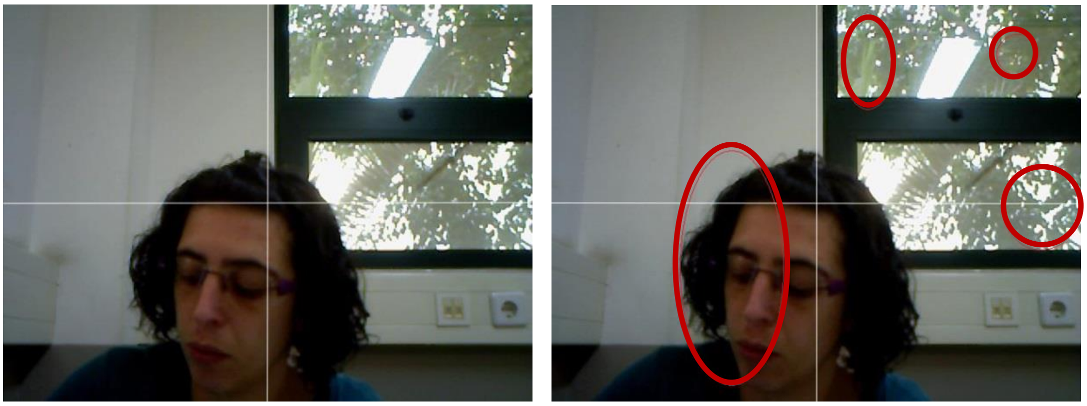

Figure 3.3 Pictures taken by the application, when the user was present, with half second of interval, the right picture contains red indications of movement compared with image on left

From visual inspection of the pictures taken with the application it was detected that the amount of pictures save it was too much. One photo per each half second was too much because there was not too much difference between pictures (see Figure 3.3). The main purpose of the pictures was to find out if the person was present and if it was detected movement by the user.

In order to improve future systems, making them more "light" and having into consideration other variables it was decided that the pictures would not be store at the computer and they should be taken with higher intervals apart. Other important enhancement should be made in the database, instead of having an online database, having a local one would bring more efficiency in the transferring and access of the information.

3.2. Research hypothesis

H1 – Can posture be predicted from distance to screen?

Why distance its important? The distance that user are from screen it's an important metric to have into account when people work daily in the computer. Even researchers do not have an agreement into a minimum distance from screen; ones say that greater than 39.9cm [22] it's good, while others say that it should be between 60 and 90cm [21].

Predicting the user posture using only the distance from scree to user can tell us a lot about it. If people reflect a little bit about their own posture when they sit in front of the computer they will realize that they can translate it in to a distance from screen. For example if you are sit too bend in front of the screen you will be at a smaller distance from screen compared with the one that you were when you don't had the head bend.

H2 – Can stress be predicted from motion detection?

Motion detection it's important to be tracked why? Using movement detection to collect information about an environment could help to understand some people behaviours. Normally people cannot sit still for long periods. So capturing movement could help in the prediction of the user mood. On the other hand, absence of movement may tell us that there are no people in the surroundings.

Predicting stress can be something really tricky and inconstant, because people behave at different ways when they feel stress, some of them move a lot – head, fingers, feet's, etc. – and others don't move at all. Using motion detection let's see if it's possible to detected movement or absence of movement when users feel more stress.

H3 – Can involvement be predicted through the mouse and keyboard activity?

Why it should be tracked the mouse and keyboard activity? The mouse and/or the keyboard might give a reflection about how frequently the person uses the computer and how engage it is with their tasks. Different tasks sometimes involve different uses of the mouse and/or keyboard, getting track of these differences might facilitate the understanding from the level of engagement of the user.

Using mouse and keyboard to predict if the user is involved on his current task can be delicate, because people sometimes have unexpected behaviours. For example some people use the computer as support while they are studying, in this case they might not use a lot the keyboard, but probably the mouse. In general if someone is using the computer to work or to have support in his work, they will probably use the mouse or the keyboard, one more than the other.

H4 – The need for a break can be predicted through posture, stress and involvement of the user?

The necessity of a user to do a break it varies much and it's not something that all persons have in common. A lot of factors had influence on it; some of the more important factors might be associated with: how uncomfortable the person is when sat, how stressed is felling at the moment and how involved it is in task that's performing (see Figure 3.4).

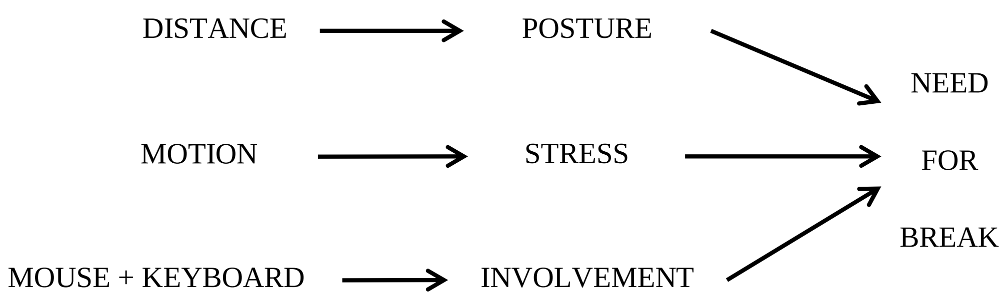

Figure 3.4 Diagram of how hypothesis relate them self

In this study we intend to infer the users viewing depth from the computer screen through face detection. We also expect to use motion detection to infer how stress the user is. In third place we aimed to log the mouse and keyboard activity to predict if the user is feeling involved with his work at the computer. The goal to predict the users need for a break it's combined the previous three predictions - posture, stress and involvement.

3.3. Research Design and Methods for Study I

3.3.1. Population, Study Sample and Sample Size

The population used in this study were students from University of Madeira, which used a Windows laptop with web camera. It was choose university students mainly because they spend more than half of their time sit either in class either studying.

This study had a sample of 4 participants and they performed the study for ten days.

3.3.2. Collection of Data

It was performed one hour observation before the user start using the application, to collect data from his posture in front of the computer. With these observations it was collect the following data: viewing depth; standard posture of the user (may be or not a good posture), how the laptop is positioned, the type of chair and table.

The application that users used collected quantitative data that was stored in a local database (see Appendix VIII - Entity–Relationship Model for Study I).

The information needed will be obtain through pictures taken with the web camera and popup questions (see Figure 3.5).

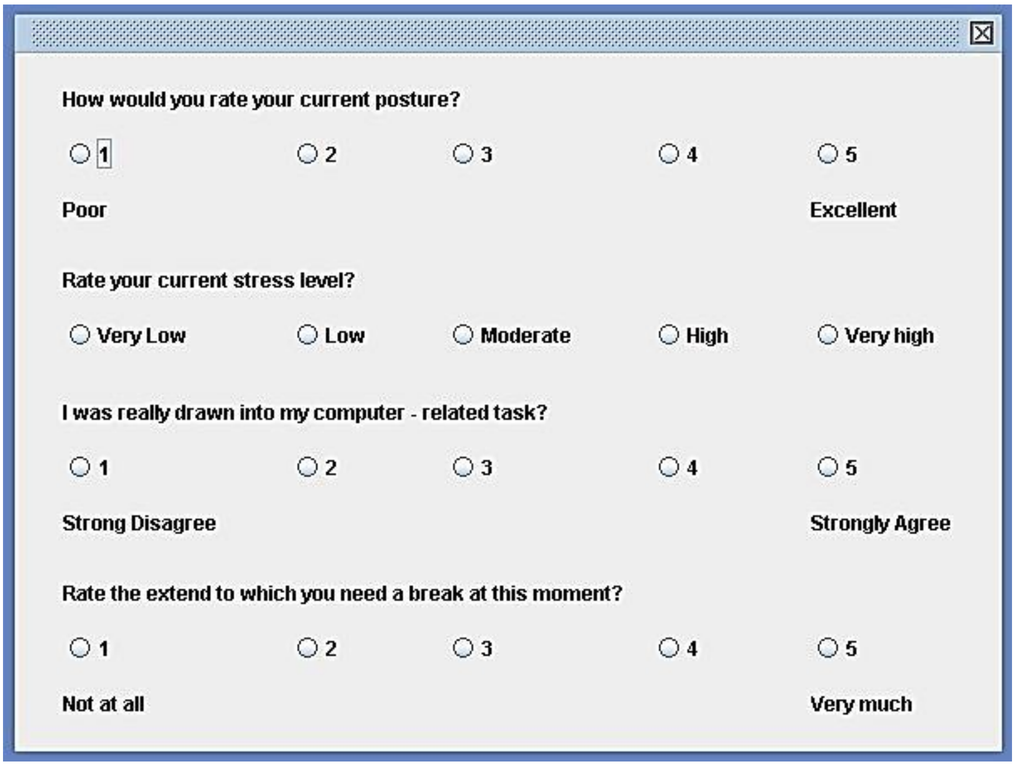

Figure 3.5 Pop up questionnaire from study application

The popup questions was shown at random times and the questions were related to the actual posture, level of stress, level of engagement and the need of a break of the participant. The answers were measured in Likert scales of 1 to 5, where 1 had a more negative connotation and 5 it was for a more positive connotation.

The pictures were taken every four seconds while the application was running following the procedure: first detection of motion, if movement present then analyse the distance moved (compared with the last picture taken) or no movement was detected; second detection of the number of persons present in front of the computer, if someone detected, gives their distance and posture in front of the screen (see Figure 3.6).

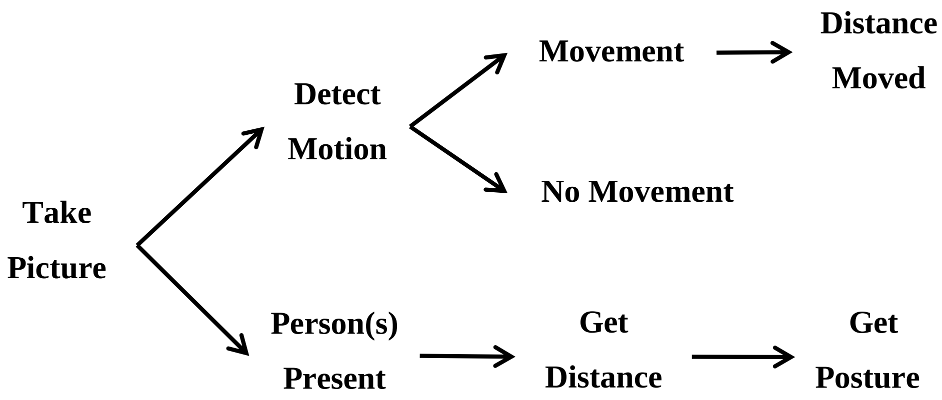

Figure 3.6 Procedure follow after take a picture

Every thirty seconds it was evaluated the mean, median and ratio for the distance of the movement detected and for the approximate distance of the participant to the screen, only for the last thirty seconds.

It was attempted to get answers to the following question: Can posture be predicted from distance to screen? Can stress be predicted from motion detection? Can involvement be predicted through the mouse and keyboard activity? The need for a break can be predicted through posture, stress and involvement of the user?

3.3.3. Data Analysis Strategies

Before starting the analysis it was needed to execute a clean-up of the data, because the program may detect wrongly some metrics.

After the data cleaning we will start analysing the collected data and we will need to analyse statistical significance tests, correlation analysis and linear regressions to give a better support to the veracity or not of the hypothesis. In some cases it was needed to achieve measurements of central tendency and variability such as: mean, median and standard deviation to complement some of the finding.

3.3.4. Ethics and Human Subjects Issues

In this research we may encounter two main issues, one is related to the use of the web camera to take pictures and the second is related to the installation of some extra applications/framework for the study application could work properly.

The web camera it's only use to take pictures but these ones aren't save to the computer, they are only to do a better analysis of some functions related with presence, posture and movement. Participants may still feel "too observe" and have a different behaviour than the normal one.

For the need on the installation of extra applications/frameworks people may be unwilling to have more things in their computer because they won't need then for their normal work.

3.4. Reused work from the Pilot Study

From the Pilot Study one of the main improvements its related with the database, it was needed to perform this changes because calls to an online database make the application work slowly and dependent on the internet connection. In order to reduce the time of this calls and the problem when no internet connection was detected, it was created a local database[2].

Duo to changes made in the database and the connection (see Appendix VII - Entity–Relationship Model for Pilot Study and Appendix VIII - Entity–Relationship Model for Study I) it was needed to perform some changes in the code that collect information to be stored in the database.

Based on observations on the results from Pilot Study it was need to perform some alterations in the code to detect motion (increase the time to take each picture and remove the pictures storing). Since it was important identify clearly a person in the pictures taken, it was added the face detection to that purpose (describe in section 3.5.1, page 34).

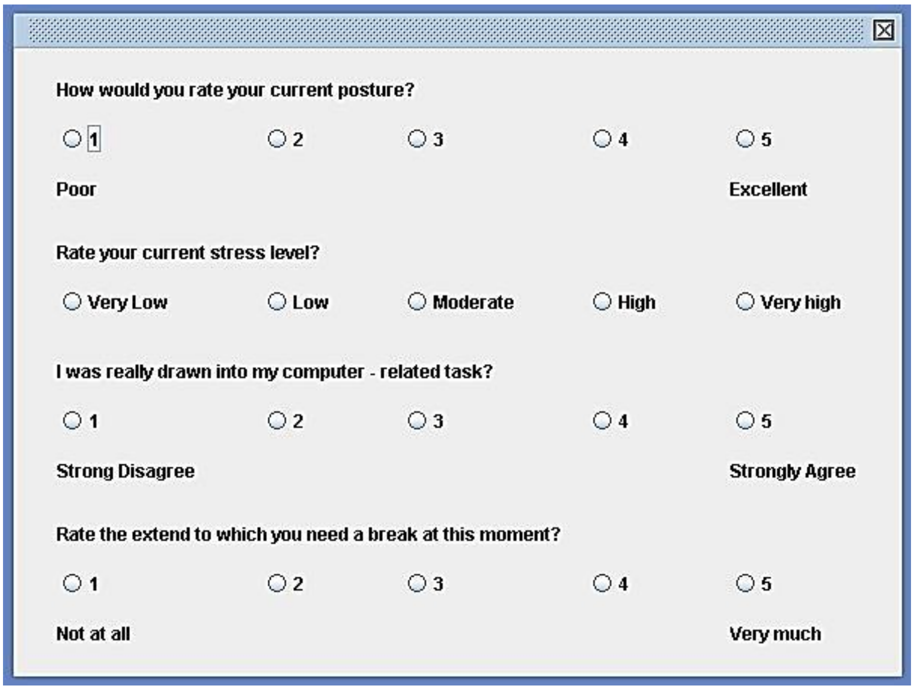

Figure 3.7 Study I popup questionnaire

The second main change was in the Popup questionnaire (see Figure 3.7), the main structure was the same and some bugs were corrected. The questions made were: “How would you rate your current posture?”, “Rate your current stress level?”, “I was really drawn into my computer – related task?” and “Rate the extend to witch you need a break at this moment?” All

[2] Local database created in Xampp an Apache web server

answers were measured in a Likert scale of 1 to 5 where 1 had a weak connotation and 5 a strong connotation.

As in the Pilot Study the information obtained with the application was shown to the user, and they liked to have this kind of knowledge, in this version it was included information related to the face detection, the analysis of the distance and for how long it was detected absence of movement (see Figure 3.8).

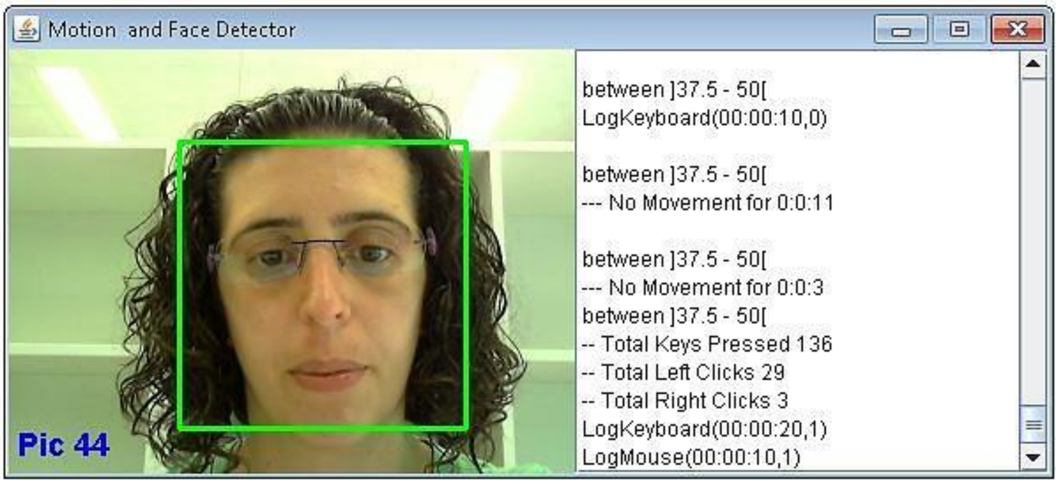

Figure 3.8 What users saw when used the application in Study I

3.5. Predicting Posture

Finding at least one measurement that could predict posture it's one of the main purposes of this study. To get answer to this problem it's intended to obtain response to the question of H1 – Can posture be predicted from distance to screen?

To predict posture, it was used a face detection approach combined with the distance that the participant was to the computer screen.

3.5.1. Face detection

The face detection was introduce in this new version in two separate ways, the first one its related with all pictures taken in the app and the second one with the main settings to detect the distance between the screen and the person.

To detect a face it's was used a java class created by Andrew Davison as base (update of March 2011; face detection class taken from [23]). This class suffer some changes in order to obtain the detection of faces correctly and it was added new functions to obtain more information.

To the original face detection code it was added a square in the location of the face, but for this approach it was needed to know not if a face was found but also the dimensions of the square, so it was made some alterations in Davison code in order to store this information.

With the original code of the face detection it was possible to know how many persons were being detected, but for the approach taken with this study that was not enough. Since the number of people present was identified through squares, it was used that information to collect posture metrics. Once the squares were not accessible to user and they would have a fundamental role in several moments, for example when detecting the Settings pictures, it was added that functionality to the original code. Based on the squares sizes it was collected information about their height, which would be used to detect distance between the user and the screen (describe in section 3.5.2, page 36).

Some of the new additional features added to Davison file were: choose the type of Haar Classifier Cascade file to be used (these files have measurements to identify frontal and profile faces), flip an image, analyse the distance (describe in section 3.5.2 page 36) and identifying the interval that belong the distance (describe in section 3.5.2 page 36). The first step performed with these new functions it's related with the type of detection that it's intended to be done. It was chosen only two types of identification, the frontal face detection (see Figure 3.9) and the left profile detection (see Figure 3.10) because after some tests they seemed to be the most accurate for the type of detection desired.

Figure 3.9 On the left side frontal face detection and on the right side the original picture

The process always begins with the frontal face detection; if no face it's detected, the left profile it's performed. If no face it's detected again it's executed the right profile detection, but before doing this last verification it's needed to flip the image in the horizontal (see Figure 3.11). This last step must be made because the profile detection available was only developed for left detections.

Figure 3.10 On the left side the Profile detection and on the right side the original picture

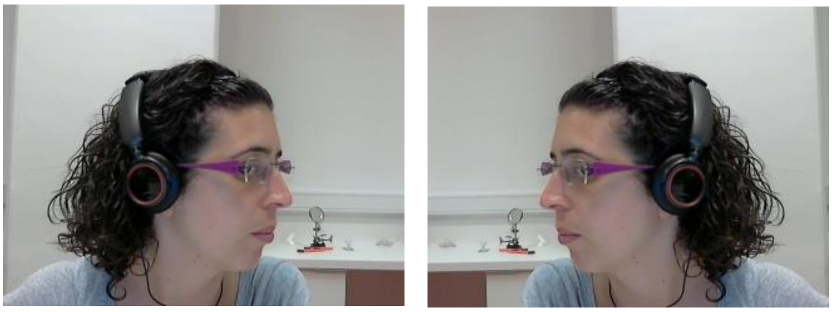

Figure 3.11 On the left a picture with right profile and on the right side the horizontal flip

After the detection completed it's needed to be done a new horizontal flip (see Figure 3.12) to have the original picture with the profile identification in the correct place.

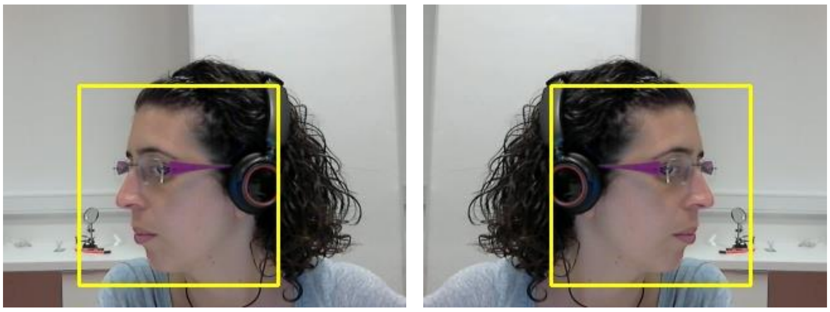

Figure 3.12 On the left side the Profile detection of the flipped picture and on the right side the original picture after the new flip

3.5.2. Distance detection

To detect the distance that a person is in front of the computer it was used the face detection combined with real distance measurements. To incorporate distance measures in the program, it was measured the distance between the subject and screen. These measurements were made with a metric tape at specific distances. The chosen distances were 25cm, 50cm, 75cm and 100cm (see Figure 3.13). It was chosen this set of distances having into account if it's possible to work at the computer in that range. It was consider that people could not work properly being at less than 25cm and at more than 100cm of the screen, as it could be perceived in the Figure 3.13.

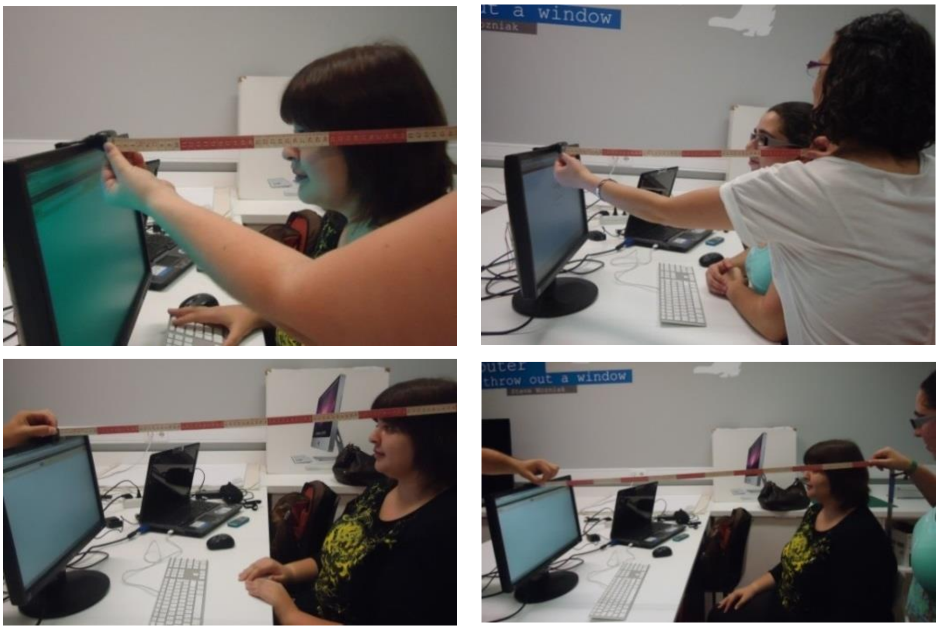

Figure 3.13 Measurements collected at 25cm in the upper left side, 50cm in the upper right, 75cm in bottom left and 100cm at bottom right

The process follow to collect these measurements was: take ten pictures with frontal face detection to each distance. Based on the size of the rectangles that identify a face its “known” which set of distance belong. The closer the user is in from the screen the higher is the square from the face detection and the farther the user is, the smaller it’s the square. For the user know exactly with distance he must be positioned the application shows a popup letting he now with distance he must be located (see Figure 3.14) after he is positioned at the correct distance to continue he must click OK. This process was repeated for the next distances; after all measurements were taken it was required to identify the settings to be used for each measurement.

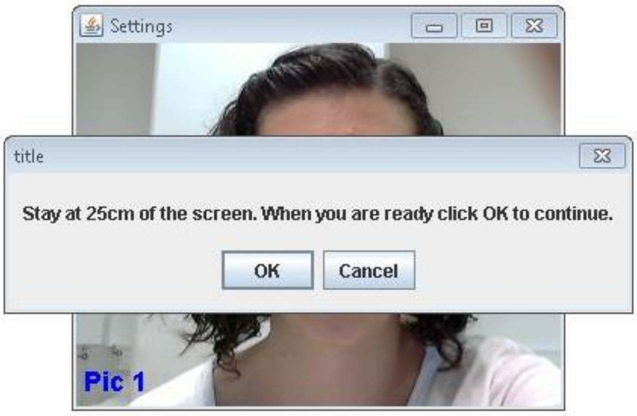

Figure 3.14 Popup identifying the first distance that the user must be at the current moment

The settings used to identify each distance were related with the size of the squares that identify a face. The setting for the 25cm distance it’s the average of the dimensions of all squares from the pictures taken at 25cm, the same process was apply for the next three distances. When concluded the settings finding, the data was saved in the database and it was used in the main application to identify the distances for each picture taken. To make the collection of the measurements a single event all pictures taken were saved in the following structure of folders see Figure 3.15 created in the same location than the main application.

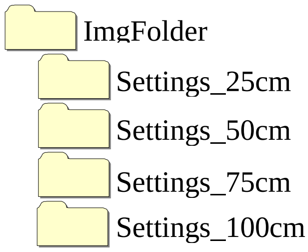

Figure 3.15 Folders structure where the settings pictures were save

To make the application the less disruptive to the user each time the application it was open it detected if the upper structure of folders exist and had ten pictures each. If this structure existed, each picture was processed to obtain the rectangles that detected the face and it was rebuilt the settings for each distance; if not it was repeated the process already described.

In the face detection class it was created two functions that relate each other, **Analyse the Distance** and **Identifying the Interval that belong the Distance**. This two functions use the settings collected with the process described above to identify the distance. The first step made after a picture taken, it's to know to which interval the dimensions of the square created belong. To verify this it was made the following set of comparisons:

- *new dimensions* > *settings 25cm*?
- *settings 25cm* >= *new dimensions* AND *settings 50cm* <= *new dimensions*?
- *settings 50cm* > *new dimensions* AND *settings 75cm* <= *new dimensions*?
- *settings 75cm* > *new dimensions* AND *settings 100cm* <= *new dimensions*?
- *settings 100cm* > *new dimensions*?

For the first and last comparison it's clear that they belong to the intervals [0 ; 25[and]100 ; +∞[cm but for the three middle comparisons it's not very clear. For this three comparisons it was apply the function for **Identifying the Interval that belong the Distance**. In this new function it was used the following procedure:

- Get the *middle* dimension base on the previous and next settings
  - if *settings next* were greater than 50cm
    - *new dimension* > *settings next* AND *new dimension* < *middle* then interval **]*setting previous* + 12,5 ; *settings next*[**

        - *new dimension* >= *middle* AND *new dimension* < *settings previous* then interval **[*settings previous* ; *settings next* – 12,5]**
    - if *settings next* were less or equal to 50cm
        - *new dimension* >= *settings next* AND *new dimension* <= *middle* then interval **[*setting previous* + 12,5 ; *settings next*]**
        - *new dimension* > *middle* AND *new dimension* < *settings previous* then interval **]*settings previous* ; *settings next* – 12,5[**

For example let's consider that the new dimension belong to and interval between settings 25 and 50 cm, so settings 25 will be *settings previous* and settings 50 will be *settings next*. Following the upper procedure the *middle* value will "decide" from one of the two possible intervals:

-]25 + 12,5 ; 50[=> **]37,5 ; 50[**
- [25 ; 50 – 12,5] =>**[25 ; 37,5]**

3.6. Predicting Stress

Stress it's the second most important metric to be obtain with this study. In order to do so, it's intended to prove the veracity of the H2 – Can stress be predicted from motion detection?

Since stress was associated with fidgeting, and fidgeting it's related with movement that people do when they are abstracted with their tasks, it was implemented the motion detection to capture this movements.

3.6.1. Motion detection

The motion detection class used in this application was developed by Andrew Davison (Motion detection class taken from [24]) (update of May 2011) and suffer minor changes. The changes performed in the original file were: exclusion of some variables, increment of the delay time between each picture, addition of a motion timer and a function to save the changes made in the pictures taken. From visual inspection of the results collected with the Pilot Study, it was more than clear that the time between each picture was too short (see section 3.1, page 26). Since it did not presented major alterations it was increased the time from 0.5 seconds to 4 seconds between pictures captured. The motion timer created, had the purpose to track for how long the application was running without capturing any movement. Capturing absence of movement was important, because it was one of the "requirements" to now if the person was at the computer or not. Before the movement detection is executed, it's performed the face detection (described in 3.5.1, page 34) and the information produced its all store in the new created function.

The movement detection, captured from each picture the points were it was detected alterations. These alterations were tracked base on comparisons between the current picture and the previous one (see Appendix VI - Process to detect Motion, for more detail about what Davidson follows to detect movement). The result of this detection was given through a coordinate Point, where it was detected higher alterations of movement.

3.7. Predicting Involvement

The third goal of this study, it's to find out a measure that could predict the user involvement when it is working at the computer. In order to achieve this goal, it's intended to prove the veracity of the H3 – Can involvement be predicted through the mouse and keyboard activity?

3.7.1. Mouse and Keyboard activity

The mouse and keyboard activity classes used in this study were developed by Kristian Kraljic (Keyboard / Mouse Hook, version 0.3, [25]). With these classes it was possible to collect the mouse and the keyboard action, without restriction to a specific window. The actions tracked from the mouse and keyboard were related to the amount of clicks (left and right) and movements made with the mouse and the number of keys that were pressed.

3.8. Behaviour analysis

The implementation made to analyse the user behaviour it's related with the faces and movement detected and it's examine every thirty seconds since the application is running.

The method followed had the purpose to obtain the average, median and respective ratios for the movement and the faces detected, as well as indications for the average and median of the correspondent interval distance in front of the screen. The procedure run was: first obtain all the information about faces and movement detected in the last thirty seconds and the day average for both detections. Second perform the appropriate maths when it's identified at least one face and movement; and in third save all the calculated information in the database.

The ratios for the face and movement detection were obtained by the following equations: Equation 3.1 and Equation 3.2:

$$FaceRatio = \frac{AVG(faces\_In\_Last\_30\sec)}{Day\_AVG(face)}$$ Equation 3.1

$$MovementRatio = \frac{AVG(movements\_In\_Last\_30\sec)}{Day\_AVG(movement)}$$ Equation 3.2

3.9. Architecture for Study I

In Figure 3.16 it's represented the architecture for the Study I, split into three levels. First level represents all the necessary equipment for the proper functioning of the system developed. The second level it's composed by all the metrics collected with the application in order to perform an analysis (third level) to obtain Posture, Stress, Involvement and Need for a Break predictors.

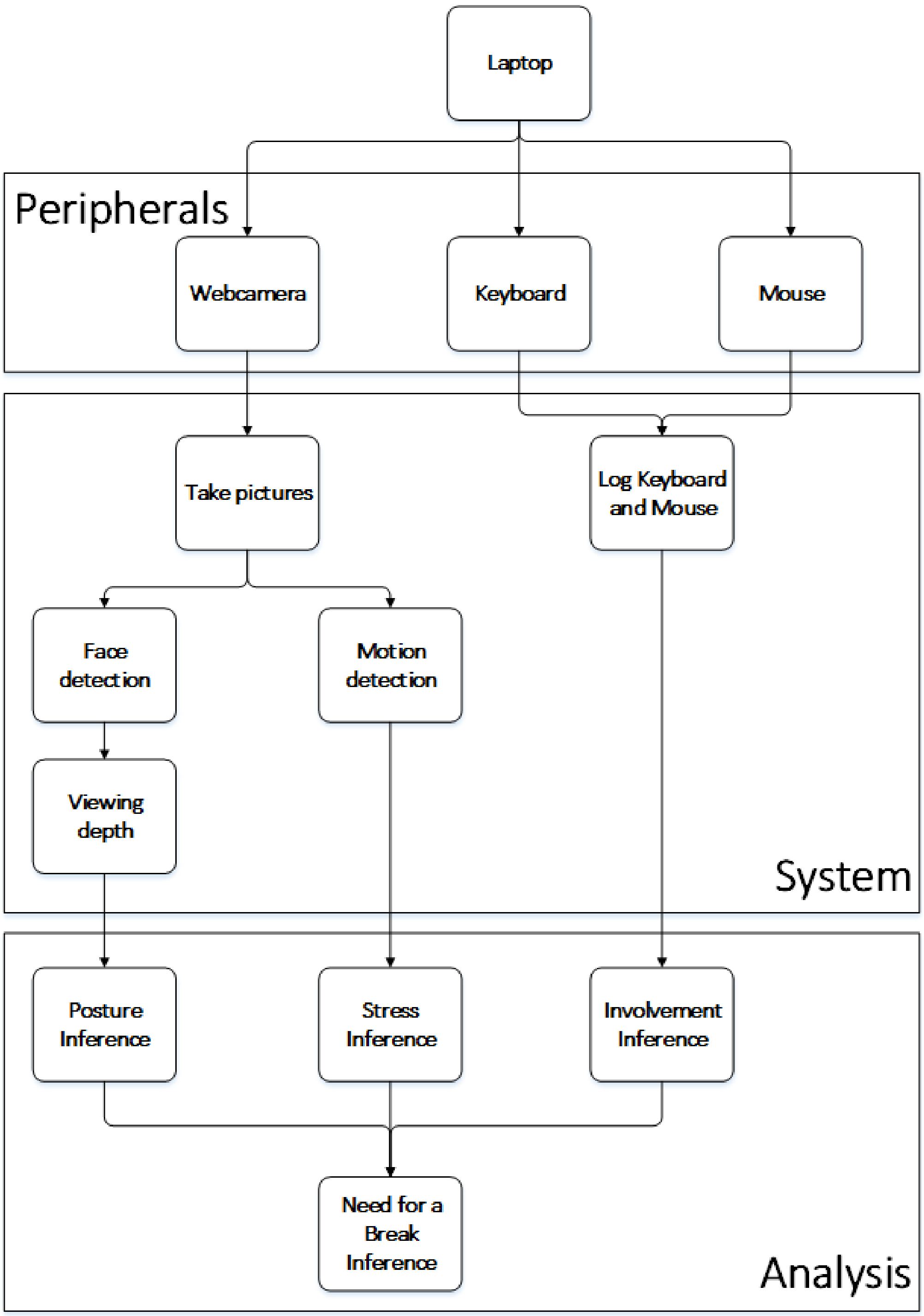

Figure 3.16 Architecture for Study I

3.10. Results

During the clean-up of the data collected by all ten participants it was noted that a bug in the popup questionnaire make mandatory the repetition of this study with an application that had this bug corrected. With the replication of the study we had to short up the study for only three days keeping the same participants.

During the clean-up of the new data we had the need to exclude one of the participants because he only use the application for less than three hours and his data won't have a strong impact in the reminding. The process used to clean-up the data had two stages: first we proceed of the removal of all popup questions that had no answer for one or more questions, second we group up by participant all data that have the same date in all tables (see Appendix IV - Entity–Relationship Model for Study I).

To verify the veracity of the hypothesis presented the first step performed was to analyse if there was correlation between the variables that we wanted to analyse, and to do that we used a Two-tailed test of significance and the Spearman correlation coefficient (the Spearman correlation coefficient was chosen because of the type of variables that were being analyse based on [26,27]).

3.10.1. Posture

The analysis made for the Posture it was related with the distance that the participant is from the screen, and to do this relation it was used the face detection that produced a square in the area with the face (see Figure 3.18 as example). The distance was represented by strings so before the analysis it was needed to convert it into a scale (see Table 3.1 variables *SI Mean HSDF - Scale of the Interval that belong the Mean HSDF* , *SI Median HSDF - Scale of the Interval that belong the Median HSDF*) the conversion was made as Figure 3.17:

Figure 3.17 Conversion of the face detection intervals in cm to a Likert scale of 1 to 5

The goal of the scale represented in Figure 3.17 it's to relate viewing depth with a scale that represented good and bad postures. It was had into consideration that experts consider a good posture when the user its sat in a range between 40 and 90 cm to computer screen.

The data collected to verify the veracity of the **H1** was the following Table 3.1:

Correlations H1		
		Posture Question Correlation Coefficient
Spearman's rho	Mean HSDF	0,417**
	Median HSDF	0,397**
	SI Mean HSDF	-0,267**
	SI Median HSDF	-0,302**
**. Correlation is significant at the 0.01 level (2-tailed).		

Table 3.1 Correlation between Face detection measurements and popup question about Posture

A positive correlation was found between posture and the mean of the height of the square that detect the face - *Mean HSDF* - (see Figure 3.18) and the median of Height of the Square that Detect a Face every thirty seconds - *Median HSDF* where the mean and median were tracked every thirty seconds. This results can be interpreted as: for higher values of the mean and median HSDF it correspond to higher values of the question about posture; and lower values of the mean and median HSDF it correspond to lower values of the question about posture. This means that when the program detects that the user is nearest to the screen the users reported being in a good posture and when it detects that the user is further the users reported being in a bad posture.

Figure 3.18 Face detection with the user away from the screen 50cm on left and 100cm right

A negative correlation was found between posture and the scale that identifies distance from the user to screen for the mean and median HSDF, these relations happens because of the scale created (see Figure 3.17). The results can be analysed as: for higher values of the SI mean and median HSDF it correspond to lower values of the question about posture; and for lower values of the SI mean and median HSDF it correspond to higher values of the question about posture. We conclude that when the program detects that the distance to the screen of the user is less adequate (appropriate) the users reported being in a good posture and when the program detects that the user in a more adequate distance (appropriate) the users reported being in a bad posture.

Based on the results obtained (see Table 3.1) for posture relationships it was verified the veracity of the H1 – Can posture be predicted from distance to screen? With these findings it was chose the relation with higher correlation concerning the height of the face detected – *Mean HSDF* – and distance from the screen that corresponds to the interval - *SI Mean HSDF*.

3.10.2. Stress

The data collected to verify the validity of the **H2** was the following Table 3.2:

Correlations H2		
		New Stress Question Correlation Coefficient
Spearman's rho	Number of movements per minute	0,239**
	Total of movements	0,150**
	Distance moved	0,068
	XCIM	0,336**
	YCIM	0,362**
**. Correlation is significant at the 0.01 level (2-tailed).		

Table 3.2 Correlation between Motion detected and Stress question

It was found correlation between the coordinates *(X coordinate that identifies the movement - XCIM*, *Y coordinate that identifies the movement - YCIM*) and the number of movements per minute.

Based on the results obtained (see Table 3.2) we accept the hypothesis H2 – Can stress be predicted from motion detection? With these findings the relation with the higher correlation was the only chose, the ordinate coordinate of the movement detected - *YCIM*. The positive sign of this value shows that when the program detects higher motion values the user reported being more stressed.

3.10.3. Involvement

The data collected to authenticate the validity of the **H3** was the following Table 3.3:

Correlations H3		
		Involvement Question Correlation Coefficient
Spearman's rho	Mouse left click	-0,386**
	Mouse right click	-0,160**
	Mouse movement	-0,428**
	Key press	-0,016
**. Correlation is significant at the 0.01 level (2-tailed).		

Table 3.3 Correlation between Mouse and Keyboard activity with Involvement question

The results in the previous table reveal weak and moderate correlations between the variables: mouse left and right clicks and mouse movements with the involvement question. It also shows that there is no correlation between the number of keys pressed and the involvement question.

Based on the results obtained (see Table 3.3) we accept the hypothesis H3 – Can involvement be predicted through the mouse and keyboard activity?

3.10.4. Need of a break

The data collected to authenticate the validity of the **H4** was the following Table 3.4:

Correlations H4		
		Need Break Question Correlation Coefficient
Spearman's rho	Posture Question	0,559**
	New Stress Question	0,733**
	Involvement Question	-0,042
**. Correlation is significant at the 0.01 level (2-tailed).		

Table 3.4 Correlation between all four questions

It was found a strong correlation relation between two questions, the posture and the stress, with the question need of a break, but there was no relation between the involvement and the need of a break.

Based on the results obtained (see Table 3.4) we refuse the hypothesis H4 – The need for a break can be predicted through posture, stress and involvement of the user?

With the results present in Table 3.4 it can be seen that the Need of a Break could be predicted just with **Posture** and **Stress** into a formula (see Equation 3.3). Since the stress predictor collected with the application was not in a scale, and to create the linear regression all variables needed to be in the same measurement. It was used the information presented in the Table 3.5 and Figure 3.19 - created with the metrics with higher numeric values of the correlations from posture and stress predictors - to create the stress scale - *Scale Y coordinate that identifies the movement (SYCMY).*

Correlation		
		Stress predictor Correlation Coefficient
Spearman's rho	Posture Predictor	0,512**
**. Correlation is significant at the 0.01 level (2-tailed).		

Table 3.5 Correlation between Posture and Stress predictors

The new scale for the stress predictor was created having into account that normally people that feel stress tend to move more but slight movement, so relating the height of the square that detected a face with the *YCIM* (see Figure 3.19) when we have more occurrences of “Y coordinate” we have more “small” movements. It was used a Likert scale of 5 levels, where 1 represent the small number of occurrences (in Figure 3.19 the white circle) and 5 it’s a lot of occurrences (in Figure 3.19 the dark blues circles).

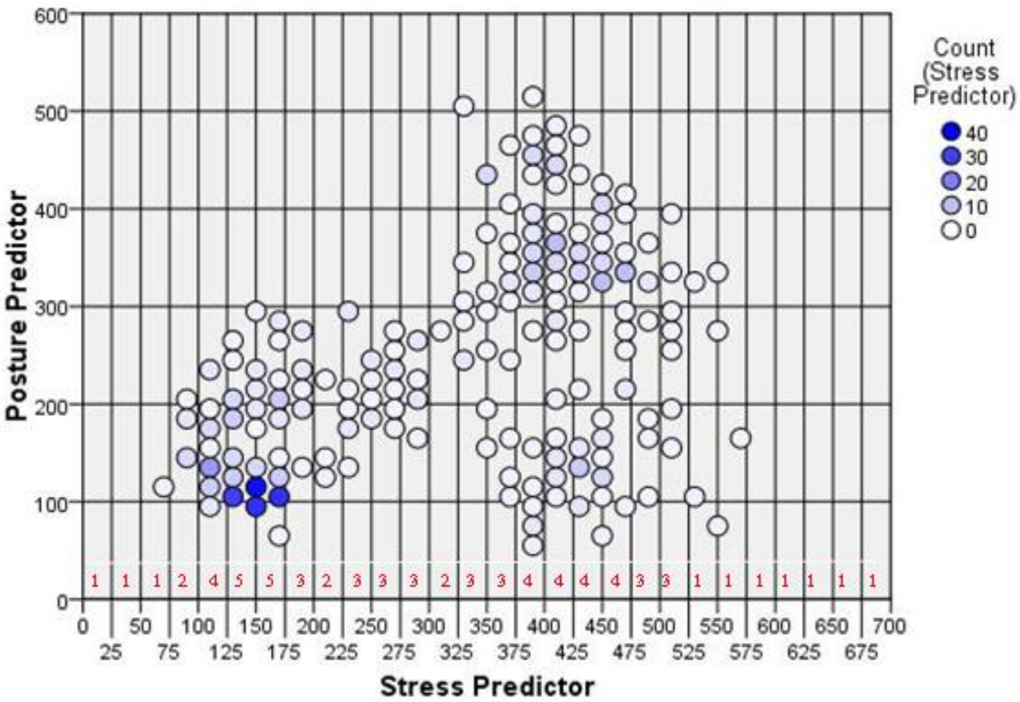

Figure 3.19 Stress scale in red based in Stress versus Posture predictors

The following Equation 3.3 was obtained through a Linear Regression where the ***Need_for_a_Break*** it's the dependent variable and ***P*** and ***S*** are the predictors. The dependent variable it's the results obtain from the need a break question measure in a Likert scale of 1 to 5.

$$Need\_for\_a\_Break = 5.590 - 0.578P - 0.190S \quad \text{Equation 3.3}$$

The predictor variable ***P*** it's the posture predictor – *SI Mean HSDF* **-** and the ***S*** it's the stress predictor – *SYCIM* – both this variables were converted to scales that were measured in a Likert scales from 1 to 5.

3.11. Conclusion

This study had the purpose to find out if it was possible to get answer to the following questions: Can posture be predicted from distance to screen? Can stress be predicted from motion detection? Can involvement be predicted through the mouse and keyboard activity? The need for a break can be predicted through posture, stress and involvement of the user? Success was achieved by getting positive answers for three of the four hypotheses. The fourth hypothesis had to be refused because one of their predictors was false, the involvement. With the analysis it was clear that it was enough to predict the need of a break just using posture and stress predictors.

These findings will have a great impact in the future application because it will be possible to correct some features and implement the new findings to improve the overall behaviour of the application.

4. System – Breaking sedentary behaviour through feedback

The system Break sedentary behaviour through feedback monitors users' levels of posture, stress and involvement at the computer tasks and attempts to motivate users to break sedentary behaviour through the use of three mechanisms: implicit, just-in-time and ambient feedback.

It was developed four types of feedback, ones more visible to the users and their surrounding colleagues and others more privates. The more visible prototypes had the goal to provide some influence in the surrounding co-workers, to encourage them too to do more breaks. On the other hand the more private ways of feedback had the aim to distinguish which would produce more impact on users, when present with a break recommendation.

4.1. Planning

The final application its base on the previous app developed for the Study I (see section 3, page 25) and had the purpose to help people to be more conscience about their behaviour when they are using the computer (see Figure 4.1).

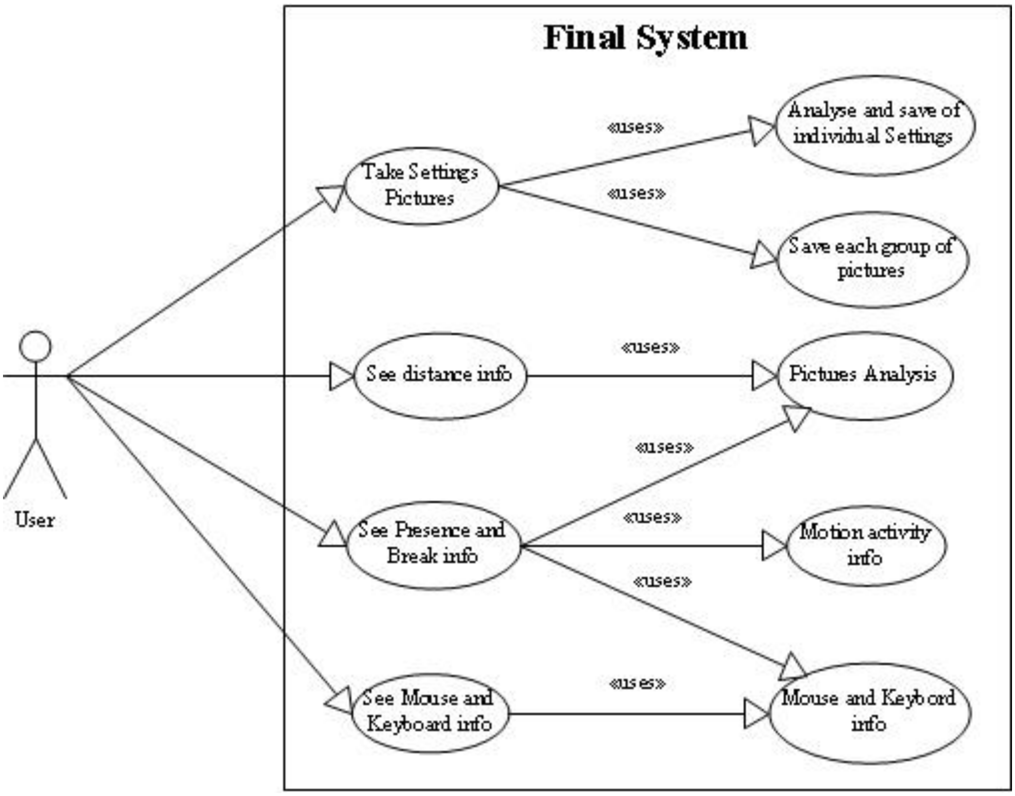

Figure 4.1 Final application Use Case Diagram

People might find in this program evidence about: their current posture base on the distance that they are of the screen, information about the number of mouse clicks, keys pressed and about their presence or absence at the computer.

The current posture of the user it's related with a good/bad posture for a person to work in a laptop (with and without external monitor and/or keyboard) without distressing the body muscles.

This application it's very simple to use and the user does not need to act differently of his normal/natural behaviour. The only extra work that the user must do at least once its related with the pictures taken for the Settings measurements. If the user at some point change of web camera it's important to collect again the Settings measurements, because the new camera characteristics would have influence in the obtained data. In this specific cases the user will need a measurement equipment (for example metric tape) to be able to position himself at the more correct distance of the screen. During the use of the application the user should pay attention to the web camera, it's important to see if the web camera it's in a good angle (and in a centre position of the screen – when using an external web camera) to capture a person face.

4.2. Hardware description

In the final application it was need to use equipment beyond the computer and a web camera as in the studies developed. It was important to create prototypes that can work as auxiliary in a way that they would improve the user behaviour. To build these prototypes it was needed several types of equipment, electronic and non-electronic, and specific tools to help mount all together.

4.2.1. Tools

To transform the sculpture prototypes into more resistant and more smalls pieces it was needed to weld the LEDs to the UTP individual wires and the resistances, for that it was used a soldering iron machine and solder.

To connect the LEDs linked to the wires and the resistances into the acrylic squares it was used a hot glue gun.

4.2.2. Construction

During the development of the final prototype it was important to have an external way to give feedback to the user. Having this in mind it was ask to student of Interface and Interaction Design - IID to start developing some of their ideas. From all the ideas presented, it was choose the Screen with the conductive ink and the Moving Cubes prototype.

4.2.2.1. Arduino sensors - capacity sensor and servo

A capacity sensor from the Arduino library can transform at least two Arduino pins into a sensor that can sense the electrical capacitance of the human body. Materials such as: plastic, wood, ceramic or other insulating material can be sense with this library. [28]

A servo motor (see Figure 4.2) it's a device that can be controlled to: move to a specific position and to move at a specific velocity. The rotation of the motor can be done between the range 0º to 180º, for example in Arduino its use the following instruction *myservo.write(90)* to rotate the servo to a 90º position. To control the speed of the motor it's used the time in milliseconds, the lower it's the time the faster it's the velocity and the higher it's the time the slower it is [29].

Figure 4.2 Servo Motor, adapted from [29]

4.2.2.2. Screen prototype

The Screen with conductive ink (see Figure 4.3) was built almost in the totality by the students Ana Canha, Dino Nunes and Joel Rodrigues. The developments made by these students were: the single tab, prolong tab and slid down; the first two ones were only developed for the sensor four - in Figure 4.3 it's the yellow wire.

Starting from the development made by the students it was completed the single and prolong tabs for the sensors one to three - in Figure 4.3 wires black, blue and green respectively - and the slid up.

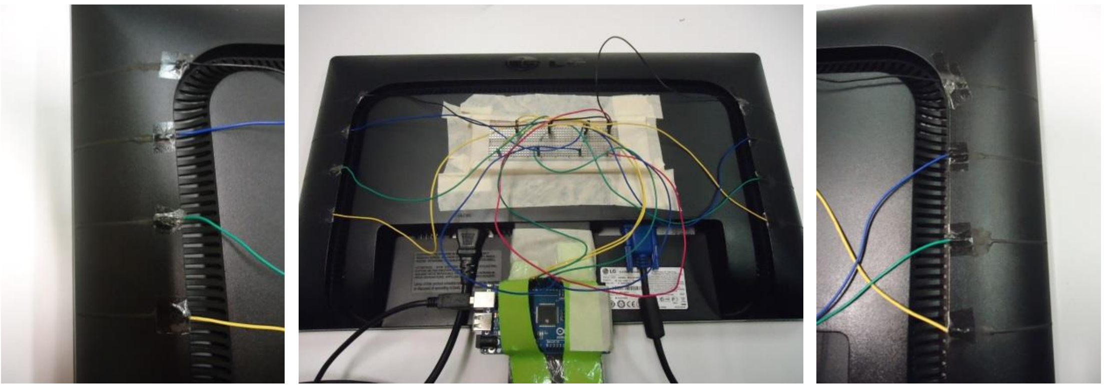

Figure 4.3 Screen prototype used

The goal of this prototype was to implement two different gestures (the behaviours of the gestures will be explain in more detail in the subsections 4.3.2 and 4.3.4), the slide down and slide up. The slide down it's the combination of the single tap in the sensors two, three and four and this movement will represent a "Yes" answer in the final application. The slide up it's the combination of the single tap in the sensors three, two and one and they will represent a delay of five minutes.

To build the two Screen prototypes it was follow the capacity sensor tutorial (code taken from [28]) and it was need the following materials: 2 screens, 2 Arduinos, 8x2 jumpers' wires, 2 USB cables, conductive ink, 4 resistances of 1MΩ and 2 breadboards (see Appendix X - Schematic from Screen prototype).

4.2.2.3. Sculpture prototype

The Sculpture prototype was built almost in full by the student Teresa Paulino. The developments made by the student were: implementation of the code to move the cubes[3], build of the structure and the cubes (see Appendix IX - How to build the origami cubes).

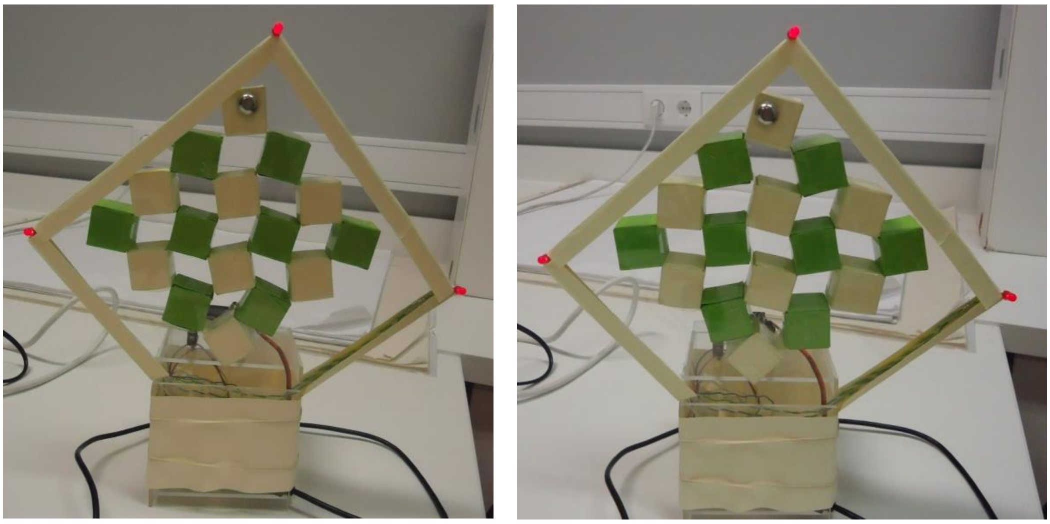

Figure 4.4 Sculpture moving and blinking when recommending a break

Starting from the development made in the code, it were done small changes, this changes were: in the delay of the servo and in from where to where the cubes were going to move. To Teresa code it was added: three led[3] that would blink at the same time that the cubes move to recommend a break (see Figure 4.4); when the cubes are not recommending a break they will move for the same position than the user posture (see Figure 4.5). Every time the application starts and ends, the cubes will be moved to a middle position (see Figure 4.6).

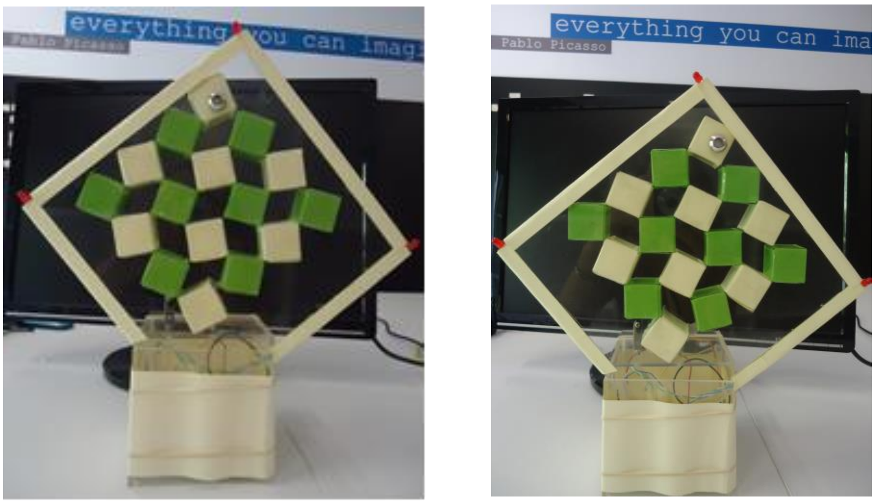

Figure 4.5 Sculpture bending forward on left and leaning back or too far position on right

[3] Code based on the example, available on the Arduino program, version 1.0.5

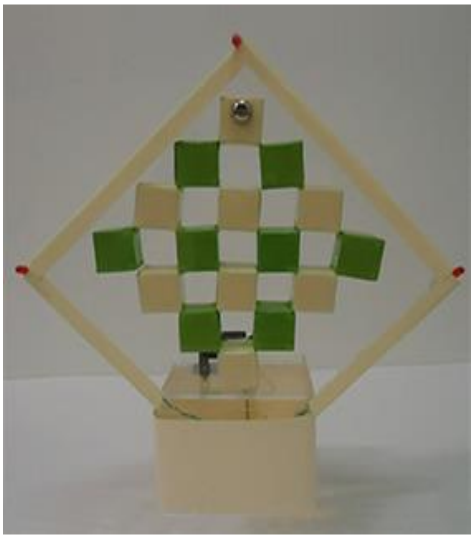

Figure 4.6 Sculpture in a middle position +/- 90 degrees

To build this prototype it was needed the following materials: 4 acrylic big squares, 6x4 acrylic small squares, 4 double magnets, 3 to 4 cardboards of two colours (green and light yellow), 2 acetate sheet, 4 Arduinos, 4 Servo motors, 4x2 supports to hold the motors, 4x4 screws and threaded, 4x2 sprockets and their fit, 4 USB cables, 4x3 jumpers wires, 2 meters of UTP cable – the needed part was the individual cables inside this cable, 4x3 red LEDs of 633nm and 4x3 resistances of 220Ω (see Appendix XI - Schematic from Sculpture prototype).

4.3. Implementation description

The development of the final application it's based on the application created for the Study I with some changes. The changes performed were: correction of bugs, the behaviour class suffers the addiction of new functions. One of the main changes was made in the behaviour class by the addiction of equations that try to predict when a person needs a break.

To show to the user when he needs a break it were created four different versions of the app. The four versions developed only change in the way the user receive feedback, they were: Popup messages combined with screen prototype, sculpture prototype alerts, combination of the two prototypes, and wallpaper changes.

To better understand how the four feedback version will create some impact in the user it was created a version were the user do not receive feedback about his behaviour and when it is recommended a break.

4.3.1. Predicting time for a break

To detect if a person needs a break it was combined the results obtain from the Study I with the time, posture and stress of the user while he is using the computer. From this study it was obtain the Equation 3.3 that relates the distance that the user is in front of the computer, plus the motion detection with a popup question about his need of a break.

To examine if user need a break it's analyse his behaviour minute by minute in the last ninety minutes to see if was detect presence or not. If the user was not present in the last ninety minutes it does not make sense to analyse his need of a break. If the user was present for ninety minutes or more it's started the checking about how long he is present, Equation 4.1; the average of how much stress the user is in front of the computer in the last five minutes, Equation 4.2 and the average of his posture in the last five minutes, Equation 4.3. The results range for the three upper equations have all values between [0 ; 1].

$$TimePersonPC = 0.3 * \frac{timePersonInPC}{120}$$

Equation 4.1

$$AvgStress = 0.3 * \frac{AVG\_Y\_last\_5\min}{AVG\_Y\_day * 1.4}$$

Equation 4.2

$$AvgPosture = 0.3 * \frac{AVG\_facedetected\_last\_5min}{AVG\_facedetected\_day * 1.4}$$ Equation 4.3

Base on the analysis made when the user it's for more than 90 minutes present it's calculated the *Symmetry*, Equation 4.4. The *Symmetry* it's the relation between the time, the stress and the posture when it's detected presence.

$$Symmetry = TimePersonPC + AvgStress + AvgPosture$$ Equation 4.4

The decision if the user its "ready" to do a break it's the combination of the Equation 4.4 and Equation 3.3. If *Symmetry* and *Need_for_a_Break* were higher than 0.45 and 2.5 respectively it's recommended to the user to do a break, otherwise he did not gather the "requirements" to do the break. In the application (only in the four different versions of feedback) is show on the *TextArea* present in the right side, information about presence and absence of presence in front of the computer (see Figure 4.7).

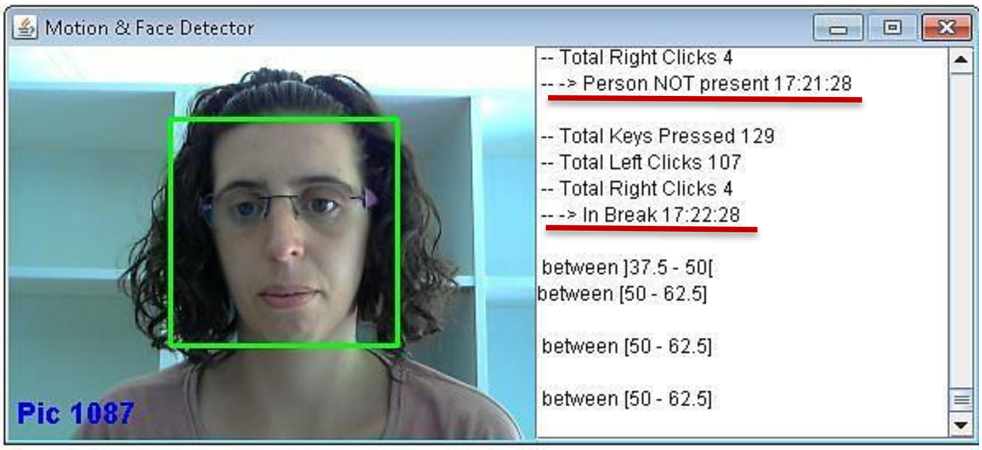

Figure 4.7 Window with information of how presence is show to the user in red

If the user is present its show the message "-> Person present for: *TimePersonInComputer* min at *hour_now*"..If the user is away for two minutes or less its show "-> Person NOT present *hour_now*" and if it's detected absence of presence for more than two minutes its show "-> In Break *hour_now*".

4.3.2. Just-in time prompting, with screen prototype

In this version of the application it's important to have the communication between the Arduino present in the screen prototype (see Figure 4.3) and the Java code. The first step to ensure this connection it's made after the detection of the Settings pictures. The second steps it's identifying from the available ports of the system the Arduino port for the screen. To help in this process, the Java code identifies the Serial port and sent to the Arduino the character "A", in this way the Arduino will know that it's the Screen prototype that is being used. The system saves the port name in order to do future connections with the prototype and closes the existent connection.

In the class that does the analysis minute by minute to detect if the user need a break its open a connection (serial port communication in Java taken from [30]) to the Screen port that remains open until the application its running. When the application detects that the user needs a break it shows a popup (see Figure 4.8) to the user that appears in the right corner of the screen. The popup exhibited (Popup code taken from [31]) does not fade out after a while and it's here that the screen prototype has his "role". To close the popup in a way of agreement the user needs to pass his hand in the conductive ink behind the screen from the Top to Bottom.

Figure 4.8 Popup show when recommend a break.

If the user do not want to be disturbed or needs a lit bit of time he must pass his hand from Bottom to Top in a way of disagreement or "asking" for a delay period. After this choice the user would not be disturbed for the next five minutes. Passed the five minutes period if the system still "believe" that the user need a break it send a new popup alert.

Figure 4.9 Popup show after the user had made delay in the first popup

In the case that the user give a positive answer if the system keeps detecting presence it will show a new popup but with a different message (see Figure 4.9) from the previous one (see Figure 4.8).

All the information generated in the tracking if the person needs a break or not was store in the local database (see Appendix XII - Entity–Relationship Model for Final application).

4.3.3. Ambient feedback

The Ambient feedback uses the sculpture prototype (see Figure 4.4) and follows the same procedure as described in the previous section (see section 4.3.2). The only two changes are: in the way of identifying the sculpture port and in the behaviour class. In order to identify the Serial port that the sculpture prototype is using its follow the same process as the one described in the previous section, but instead of sending the character "A" it's sent the character "B". The alterations made on the behaviour class were: how to suggest a break and how the sculpture will behave in the rest of the time. To activate the Arduino code for the first case its send by the Java code through the serial port the number 200 by the instruction *SculpterPort.send2bytes(200)*. After the Arduino receive this instruction it's possible to see the following behaviours: first the sculpture goes to a middle position (see Figure 4.6); second start rotating for the left; third return to the middle position; fourth rotate for the right and sixth returns to the middle position. Following the steps two to six its present the red LEDs blinking. The angles use to describe this behaviour were: 90º to 40; 40º to 90º, 90º to 140º and finally 140º to 90º (see Figure 4.4). The above procedure reflected on the state of "Prompting to take a Break". In the second case the sculpture will follow the user posture, to do this it's analyse the average of the distance detected in the last minute and apply the Equation 4.5. The *AVG(distance_lastMinute)* it's collected through the *height* of the squares for the faces detected in the last minute and the *25cm_distance* it's the *height* of the square for the average face detected at 25cm (the first Settings obtained). The above method was designated of the "Current Posture" state represented in the sculpture.

$$RotatingAngle = \frac{AVG(distance\_lastMinute) * 170^{\circ}}{25cm\_distance} \quad \text{Equation 4.5}$$

To send from the Java code for the Arduino the rotating angle of the servo it's used the instruction *SculpterPort.send2bytes(RotatingAngle).* When the sculpture receive the angle the first step its identify for with side, left or right, the new position must rotate (see Figure 4.5) after this identification, it start rotating from the last position until the actual position.

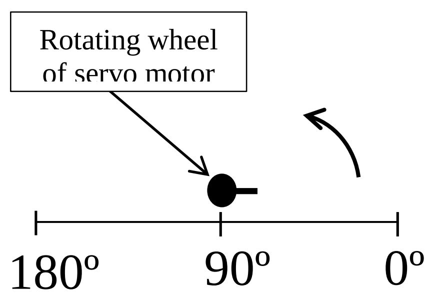

Figure 4.10 How the servo motor rotates

The *RotatingAngle* can be well perceived by the user when he uses the sculpture at his right side because the sensor motor used in the prototype rotates from the right – minimum of 0º to the left – maximum of 180º (see Figure 4.10).

4.3.4. Just-in time prompting, with screen and sculpture prototypes

The just-in time prompting, with screen and sculpture prototypes, it's a junction of the just-in time prompting, with screen prototype and the ambient feedback. All of what was describe in the sub-sections 4.3.2 and 4.3.3 did not suffer major alteration in this version of the application. The alterations made were: in the detection of the Arduino ports and in the behaviour class. In each separate version it was only needed to be detected one Arduino port and after the detection it stop the analysis, in this version the verification continues until finding the second port. In the behaviour class the change perform was in the addiction of one more serial connection to be possible communicate at the same time with the two prototypes as describe in the previous sub-sections.

4.3.5. Implicit feedback

The wallpaper version provides the implicit feedback and it's a little bit different from the prior versions (see sections 4.3.2, 4.3.3 and 4.3.4) because it does not have any Arduino component connected. In this version it was need to remove all the Arduino communications and add functions to change the current wallpaper. To change the current wallpaper (code taken from [32]) to a specific image it was had into account the time the user is in front of the computer. The four images used were all located at the path "C:\Wallpaper" in the user's computer.

Figure 4.11 Wallpaper used when the user start the application and return from a break, adapted from [33]

The code developed to switch images was added to the behaviour class in place of the older ways to recommend a break. The first image (see Figure 4.11) it's used every time the user starts the application and after his return of a break. With this image it's intended to reflect/transmit wellbeing to the user through the light colours and the animal itself.

The second image (see Figure 4.12) had the purpose to transmit to the user the feeling of the beginning of being tired. Thinking on the turtle it's know that they are more flow inside the water and when they are on land they tend to move slowly and get tired easily. This wallpaper its set when the application detects presence without breaks after a period of 60 minutes.

Figure 4.12 Wallpaper used when the user is starting feeling tired, adapted from [34]

Figure 4.13 Wallpaper used when the user is feeling very tired, adapted from [35]

To reflect a higher level of stress by the user state its set a new wallpaper image (see Figure 4.13) after a period of 105 minutes with presence in front of the computer. This image reflects a chase between two animals; normally into a chase after a while one or both of the animals begin to feel very tired. The chase image might not always appear as wallpaper to the user, if the application after a period of 90 minutes detect that the user needs a break.

Figure 4.14 Wallpaper use to recommend a break, adapted from [36]

The last state for the wallpapers images set in the user computer it’s the one that will recommend a break. The Lion image resting (see Figure 4.14) after a wallpaper image of a chase between animals had the purpose to alert the user that he might need to do a break. This picture is set based on the time that the application detects that the user needs a break (see Equation 3.3 and Equation 4.4) as describe in sub-section 4.3.1, page 58.

4.3.6. No feedback

The no feedback version had the goal of not providing any type of information as it's done in the previous types of feedback. So in this version we kept the same behaviour but we hidden it from the user. Visually the user had information only about the face detection – maintained to ensure the proper identification of users – see Figure 4.15. This functionality was kept because almost all the analysis that is made its bases in the detections made in pictures (face and motion detection) and it was important to guarantee that they were taken properly.

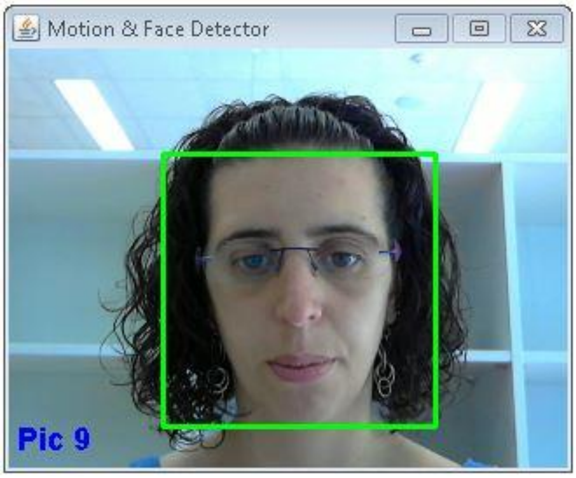

Figure 4.15 Window version of "No Feedback" application

4.4. Architecture for Study II

In Figure 4.16 it's presented the Study II architecture base on the information presented in the previous chapters. The first level it's composed by the peripherals needed for the proper functioning of the system. The second level describes what metrics need to be collected to infer Posture, Stress and Need for a Break. In the last level we have three different mechanisms to help motivate users to do more breaks.

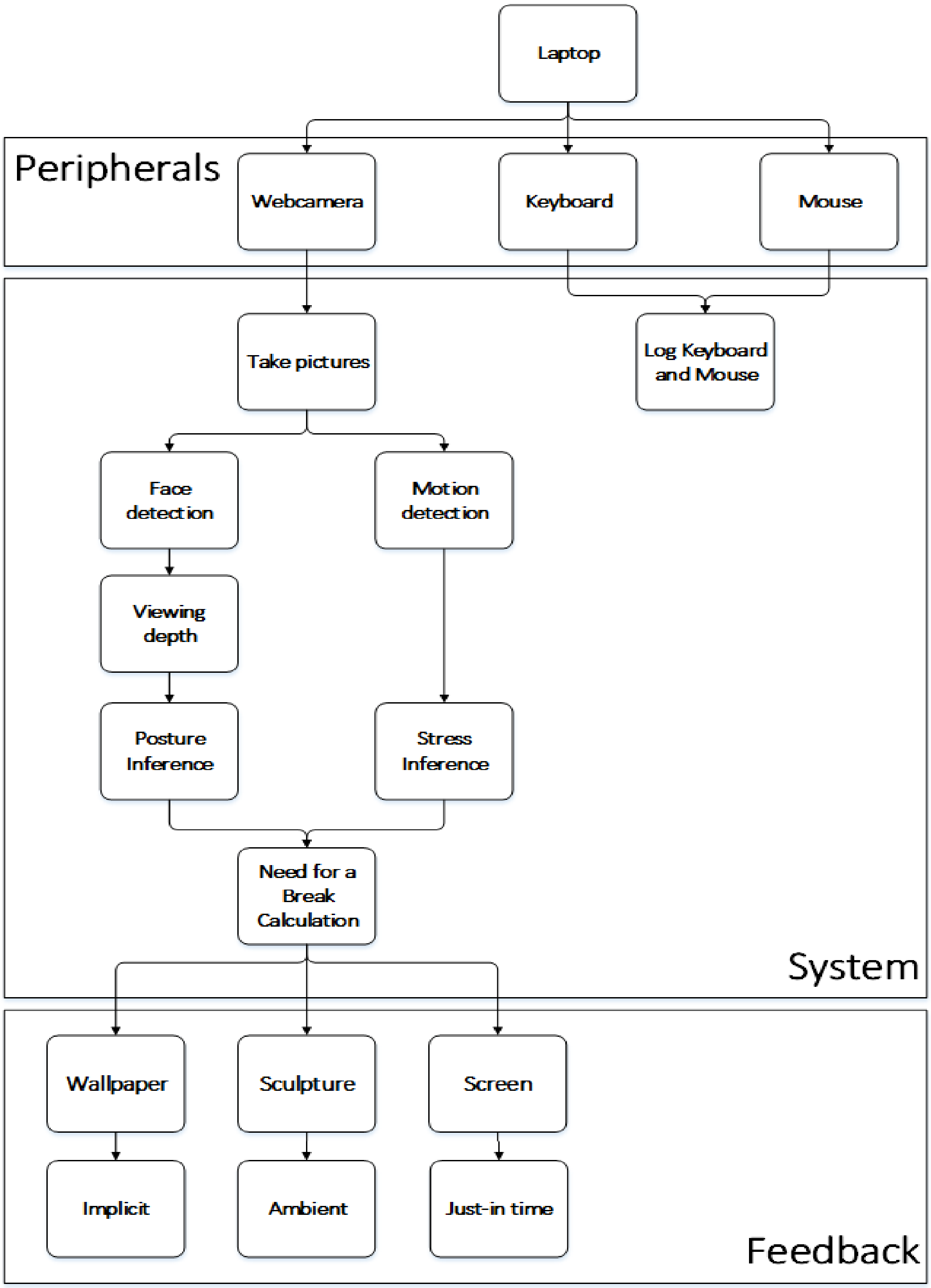

Figure 4.16 Architecture for Study II

4.5. Conclusion

In this chapter we described the design of an application that aims at breaking sedentary behaviour through feedback. We defined four types of feedback and how they were implemented using the results collected with the Study I (see section 3, page 25). Each type of feedback was associated with a different strategy: wallpaper changes, prompting alerts and visual information. In the next chapter we will describe a user study of this in a workplace.

5. Study II – Users behaviour when faced with different types of feedback

In the present chapter we report on a study that aimed to evaluate the application presented in chapter 4. The goal of the study is to assess the impact of the three different strategies of feedback on users' sedentary behaviours. In order to achieve this purpose, the users experienced four different versions of the application, three of them reflecting the different types of feedback - *implicit*, *ambient* and *just-in time* recommendations, and one reflecting the "*No feedback*" condition.

The target population – university students – used in this Study were the same as the ones of Study I (see section 3, page 25). The reason of this choice was to get consistency with the previous study, because it's a continuation of it.

In this chapter we raise the following questions: Can feedback influence sedentary behaviour? Can ambient feedback provide impact on users in order to accept a break recommendation? Can implicit feedback provide impact on users in order to accept a break recommendation? Can just-in time prompting provide impact on users in order to accept a break recommendation?

5.1. Research Questions

RQ1 - Can feedback influence sedentary behaviour?

Why feedback it's important? Feedback its important when using a program, because the user reaction after the information provided, will get influence on his/her acceptance of the recommendations made. Different people respond differently when presented with the same information, while ones might understand what they see other might not.

RQ2 - Can ambient feedback provide impact on users in order to accept a break recommendation?

Users can be influence on their choices base on their surrounding environment. Having a product that provides visual information it might be tricky, because user pay attention not only to usefulness of the product but also if it's beautiful or not [37]. It's important that the product reveal a clear interpretation of its purposes.

RQ3 - Can implicit feedback provide impact on users in order to accept a break recommendation?

Wallpapers can be implicit about their meaning? Wallpapers images reveal a lot about the user, because they are very personal. Providing pleasantness images can be difficult when thinking in people general preferences. In general people do not dislike animals, so choosing animals to represent user feelings might be a safe “bet”.

RQ4 - Can just-in time prompting provide impact on users in order to accept a break recommendation?

Prompting won’t be a problem? Users when presented with flows of alerts might feel stressed and annoyed, but when alerts are about useful information and not too often they do not break users concentration [9].

5.2. Research Method

The application Break sedentary behaviour through feedback it was used by four students, two males and two females, and it was created two different groups of testers. The users were all students from University of Madeira, three of them used the application for ten days and the fourth use it only for four days.

The groups created (see Table 5.1) were grouped following the A/B testing methodology where the “A” it’s represented by the “No Feedback” version and the “B” it’s represented by the four different versions of feedback.

Groups		**Duration**
I	**II**	**Day**
No Feedback	No Feedback	1
Screen and Sculpture	Sculpture	2-3
Sculpture	Screen and Sculpture	4-5
Wallpaper	Screen	6-7
Screen	Wallpaper	8-9
No Feedback	No Feedback	10

Table 5.1 How groups were split using the A/B methodology

In all groups the “A” variant it was used during one day at the beginning of the test and in the end. The “B” variant was used during two days for each different version of feedback except by one of the users. In this specific case the user did not concluded the A/B testing. He used the first “A” variant and three of the four versions (Screen, Sculpture and Screen and Sculpture) of the “B” variant. In this case, the participant used only for one day each version.

The A/B testing methodology was implemented to help prevent possible carryover effects, if all participants had used the same versions in the same order. The implementation shown in Table 5.1 allows us to collect rich information for each type of feedback, because they try to prevent the fatigue effects and encourage the practice effect.

After the user had used the “No Feedback” and the first Feedback version it was made an interview (see Appendix XIII - Interview protocol for the Final application), first about how the user behave daily and second about the version with feedback. For the reminding scales of the feedback versions it was done interviews at the end of each version. In the end of the tests it was ask to the user their reaction about all versions tested. It was also had into consideration the opinions of five users that share the space with the four testers. All the answers collected with the users that tested the application were identify as: **P1 to P4** and the observers were identify as: **P5 to P9**.

5.3. Findings

The results collected with the application show that the different ways of providing feedback to the users to recommend them a break have some influence, compared to when they do not have any kind of feedback. The number of total events calculated to recommend a break (see Table 5.2, column TCB) was the higher when users used the wallpaper application. The sculpture version shown the second higher results on break recommendations presented to the users. The applications that used the screen prototype to receive feedback revealed to be the version that detected more breaks done by users. The app with no feedback showed to be the one that detected the lower number of events that did not recommend a break to the user.

To the following results it was added the messages/opinions collected by the people that were observing the user when they tested each different version of the application. The total of participants and observers was of nine, all of them students, four females and five males.

The aims of this finding it's to understand if at least one of the feedback apps had more impact on the user and/or the observers and in what way.

5.3.1. Feedback techniques

The participants of this study had the chance to test three different ways of receiving feedback – *implicit*, *ambient* and *just-in time prompting* – to encourage them to do more breaks. The implicit feedback consists of changing wallpaper (see Figure 5.1); the ambient feedback consists of a desktop sculpture (see Figure 5.2) that reflects one's posture over the previous 5 minutes. Last, the just-in-time prompting consists of a traditional pop-up window that prompts the user to take a break (see Figure 5.5). Feedback it's composed by the applications that used the screen (screen version and sculpture + screen version).

5.3.2. Inferring break urgency

We used three different metrics to calculate if a break should be recommended to the user at any given moment in time. The first metric was the *consecutive-time* that the user was in front of the computer. The second was *Symmetry* as inferred from *consecutive-time*, *posture* and *stress* metrics in the last 5 minutes (see section 4.3.1, page 58). The third metric was the variable *Need-for-a-break* that we inferred from posture and stress metrics (see section 3.10.4, page 48).

We decided to prompt for breaks only after a user sat in front of the computer for 90 consecutive minutes. We did this to minimize unwanted interruptions but also as we judged that individual's need for a break is minimal below a period of 90 consecutive minutes. WorkSafeNB [21] argue that users should do exercices/stretches more less every hour, through the day. After 90 minute interval, the application started computing the *symmetry* and *need-for-a break* every fifteen minutes until a maximum of three periods (90, 105, 120). If the users reach 120 consecutives minutes in front of the computer, the program would compute the *symmetry* and *need-for-a-break* every minute. Once a user would depart from the computer to take a break, the counter would reset and the process would restart. The process to recommend a break would be restarted every time it's detected that the user is absent from the computer for more than 2 minutes.

Type of feedback	**Recommended Breaks (RB)**	**Breaks not recommended**	**Total of calculated breaks (TCB)**	**% of RB by TCB**	**Observed breaks (OB)**	**% of OB by RB**
Implicit	103	89	192	53,65	41	39,81
Ambient	73	25	98	74,49	31	42,47
Just-in time	29	147	176	16,48	73	251,72
No feedback	60	3	63	95,24	39	65,00
Total	**265**	**264**	**529**		**184**	

Table 5.2 Breaks calculated and done, grouped by type of feedback

The results presented in Table 5.2 are a brief summary of the break events – recommended and not recommended breaks and breaks detected – generated by the application grouped by the four different types of feedback (results will be explained in the next subsections).

The column Recommended Breaks (RB) show all results obtain when the users received a break recommendation. In the type "No feedback" this recommendations were not shown to users, because it was not presented any kind of feedback with this version. The column Break Not Recommended where the recommended breaks calculated that did not achieve at least one of the specifications used to consider a break (*symmetry* and *need-of-a-break*). The Total of Calculated Breaks (TCB) indicates all the events calculated to recommend a break - the recommended breaks and the not recommended breaks. The percentage of recommended breaks (RB) from the total of calculated breaks (TCB) it's shown in the fifth column. The Observed Breaks (OB) column presents all the breaks that the user really did with the application running. The last column shows the percentage of breaks really did by the ones that were recommended.

5.3.3. Implicit feedback

The majority of the breaks calculated and recommended (see Table 5.2) were detected in the “Implicit” version with 192 generated events, where 53,65% of the recommended breaks were shown to users. This fact might be because this version was the most utilized by the participants, with 24.36% of use compared with the other versions (see Figure 5.4).

A curious fact of this applications it’s that the feedback it’s not visible as in the other versions. Interestedly this was not an obstacle for users do not see the wallpapers changing; they used the Windows Aero Theme to help them. This theme has windows transparency so it’s more visible the changes at least in the task bar and it’s not need to leave an “empty space” to see the desktop.

P2 – “(...) after change to Windows Aero Theme it was more noticeable the wallpaper alteration (...) especially in the task bar”.

Although the data indicates that more than half of the breaks were shown to users, not all understood how the wallpapers behaved and what they intended to represent.

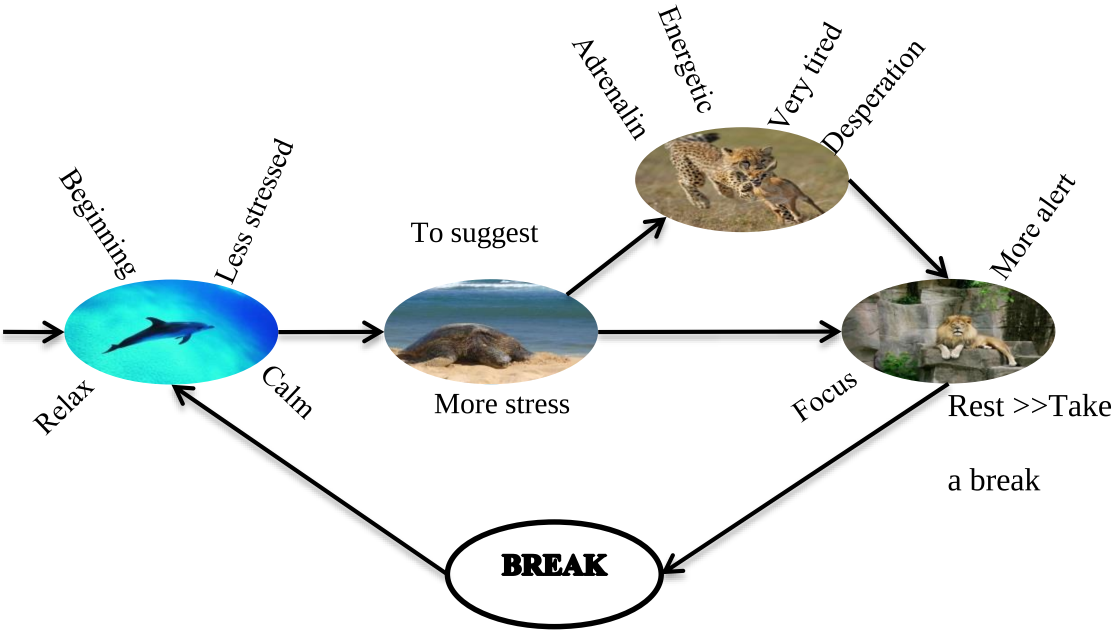

Figure 5.1 Users feeling for the different wallpapers

Figure 5.1 highlights the transition between the wallpapers as the states of sedentary behaviour increases. As one may note, each animal represented a different state of stress, the dolphin represented the lower level and the cheetah/gazelle the higher one. The lion wallpaper did not have the purpose of represent a stress level but to recommend a break. Next to each wallpaper, we highlight users’ interpretations, which often varied from our design hypotheses.

The first wallpaper presented to users was the Dolphin and in this case there was an understanding that this image represented the beginning and a "fresh" start after a break and/or the opening of the application. *P4 – "I took some time to understand the meaning, but then I realized that it was the starting point (...)"*.

After ninety minutes had passed since the application was open or the user had return from a break the Dolphin wallpaper was replaced with the Turtle. This wallpaper caused both positive and negative interpretations of what it was demonstrating. Firstly all users agree that this image characterized a more stressed level than the previous one. *P3 – "(...) begin to feel tired (...)"*. Secondly it caused a participant to interpret the image as the one which recommended a break. *P2 – "(...) associated it with tiredness and taking a break (...) every time I saw a turtle I had a break."*

If the user was at the computer for at least one hundred and five minutes and the application did not detect that he needed a break the Cheetah/gazelle wallpaper would replace the Turtle image. The first time this wallpaper appeared the users felt a little confused, because of the existence of the two animals that represented opposite feelings, although the image represents a chase and that involves higher levels of stress and concentration. The cheetah was identified by some users as being the less stressed animal and the more energetic/active. *P4 – "(...) thought that cheetah was too energetic." P5 – "(...) the expectation of reaching its goal (...)." P6 – "(...) strong and fast animal (...)."* On the other side the gazelle was expressing the fatigue, the desperation and the tiredness. *P5 – "(...) tortured by the chase (...) and frightened (...)." P6- "(...) it's going to quit at any moment (...)."*

The Lion wallpaper can be activated at any time if the user is at the computer for more than ninety minutes. At this point, what will determine its activation is the level of stress and posture of the user. The users did not have difficulties to understand this wallpaper because after seeing wallpapers that represented different levels of stress, seeing an image that showed the opposite, it was easy to interpret. *P8 – "Relaxing position but watchful eye (...)." P2 – "(...) no concerns (...) and quiet (...)." P8 – "To take a break urgently otherwise the lion would eat me."*

Two of the four wallpapers chosen were easier to understand (dolphin and lion) and the other two were the trickiest ones, probably because the images were not the most clear for the states that they intended to represent.

5.3.4. Ambient feedback

From the three ways of feedback the ambient was the one with higher percentage of break recommended with 74,49% of the events, where the real breaks detected (observed breaks in Table 5.2) with the application were only 42.47% of the events.

This type of feedback used an extra piece where the sculpture mimicked the user's current posture (see Figure 5.2) and where it was shown the break recommendation.

Figure 5.2 Participant working while using Sculpture prototype

The behaviour done with the sculpture to replicate the user behaviour was not well interpreted by all users. One of the users did not understand what the movements that the sculpture was doing when it was on the "current posture" state – in this state the LED light would not blink. *P1 – "(...) looked to see when it moved, but did not understand what posture represented. (...) I see it all 'twisted' and I thought if my spin was like that." P7 –"(...) when I see it moving I thought it was to alert to do a break. (...) I wander if the sculpture could be deactivated to stop moving all the time (...)."* One of the problems for these interpretations might be related with the choice of the movements, perhaps if it was chose other type of movement to mimic the user posture it would be clear what the piece was representing.

The sculpture blinking provoked curiosity in the user and in the surrounding persons because of the combination of the movements and/or the light bilking. *P3 – "it drew immediate attention because it started moving more than usual (...)." P2 – "Noticed some people looking to see what the sculpture was doing." P4 – "(...) in the breaks I made the association by the light and not the movement".*

5.3.5. Just-in time recommendations

The just-in time feedback from all types was the second one with more calculated breaks by the applications that compose it – screen and sculpture + screen prototypes. But surprisingly it was the one that had the smallest percentage of break recommendations (see Figure 5.3) with 16% (29 out of 176) of the calculated total and the higher number of breaks not recommended to users (see Table 5.2).

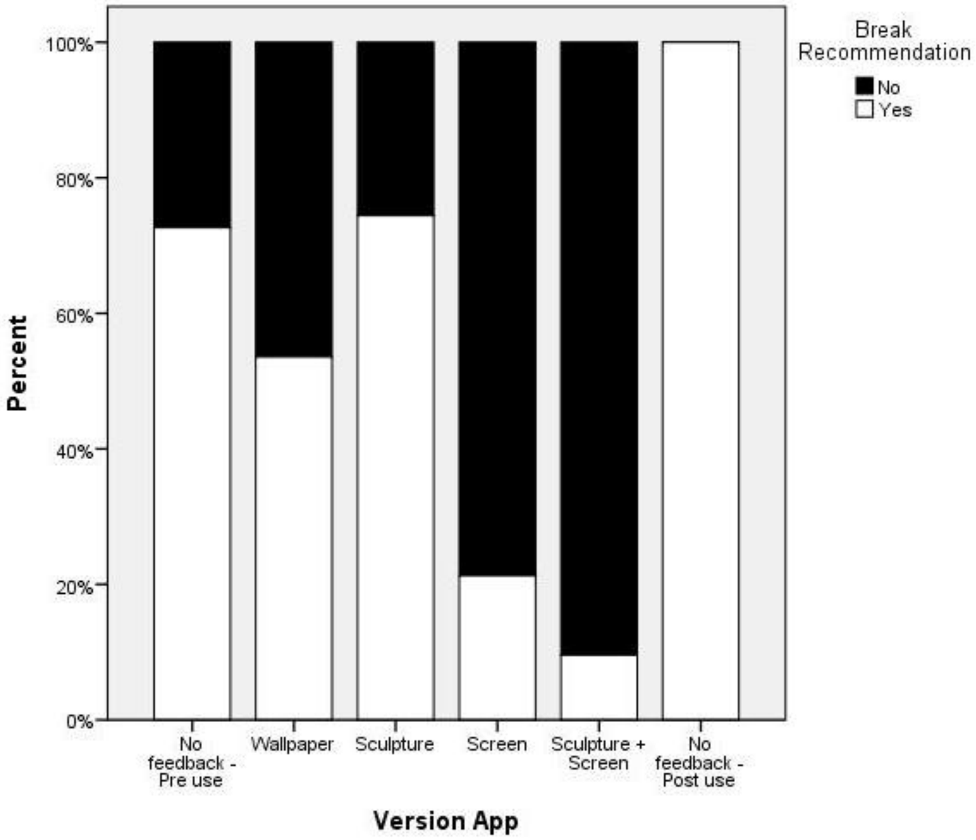

Figure 5.3 Breaks recommended and not recommended to users in each version of the application

These results might be due to several reasons, one the fact that this group it's composed by two applications that do not had the same time of use; second having two types of popup alerts, third the prototype with the screen and sculpture might be affecting the results and fourth because the number of real breaks detected was 251,72% from the recommended ones.

The use of the screen in two different prototypes when the prototypes were not use at the same time and at different orders for the users (see Table 5.1) might produce prior knowledge when they had already use one of the feedback component making them use more or less the application. *P4 – "(...) I missed the sculpture because I was already used to see my posture and correct it if needed (...)".* In Figure 5.4 it can be seen that sculpture + screen app was the second most used application – application that was running for more time -, comparing these results with the ones presented in Figure 5.3 it was expected that the number of breaks recommendations should be higher but more than 80% of the results show the opposite. The result showing more "No" break recommendations might be because the users spend less time in front of the computer and/or they weren't at a good posture in most of the time.

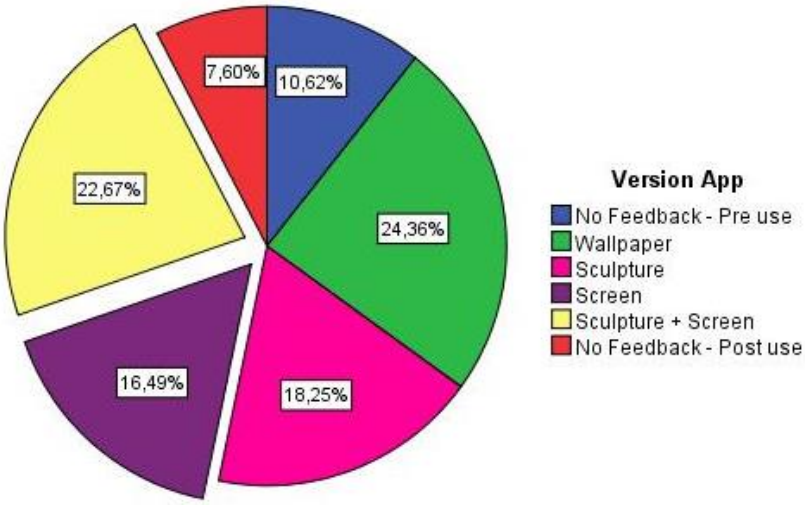

Figure 5.4 Percentage of utilization of each version, with interactive prototypes highlighted

P2 – "(…) once I noticed that I was more or less 90 minutes at the computer and I was waiting to see something show up, but nothing happened." P3 – "I stayed away from the computer a lot (…) and I got tired of waiting for the notification then I leave to do a break without recommendation (…)".

The observed breaks reveal that user where very often away from the computer with 251.47 % of breaks detected based on the recommended breaks.

The distances analysed by the program in the two versions that composed the just-in time feedback show that the average distance in prototype screen belong to the interval]62.5 ; 87.5] cm and in the version sculpture + screen to the interval [75 ; 100] cm. In the first prototype this interval fits in to a not good posture but in the second prototype it's definitely from a not good to bad posture. *P1 – "I noticed that my posture was not the best, sometimes I was in a forward leaning position and in other cases in a more relaxed posture." P2 – "(...) the distance from screen was higher than usual, and I believed that I was in a worse posture (…).*

The screen prototype had two types of popups of feedback (see Figure 5.5), the first popup it was show when the application detected that the user needed a break and the second one it was show after the user had received the first popup and "accepted" – performed a slide movement from top of the screen to bottom – but remain at the computer. Users felt more clarify that the application was recommending doing a break.

P4 – "(...) I noticed that after a while the alert show up again with a different message. (...) when it said 'REALLY' need a break I was immediately".

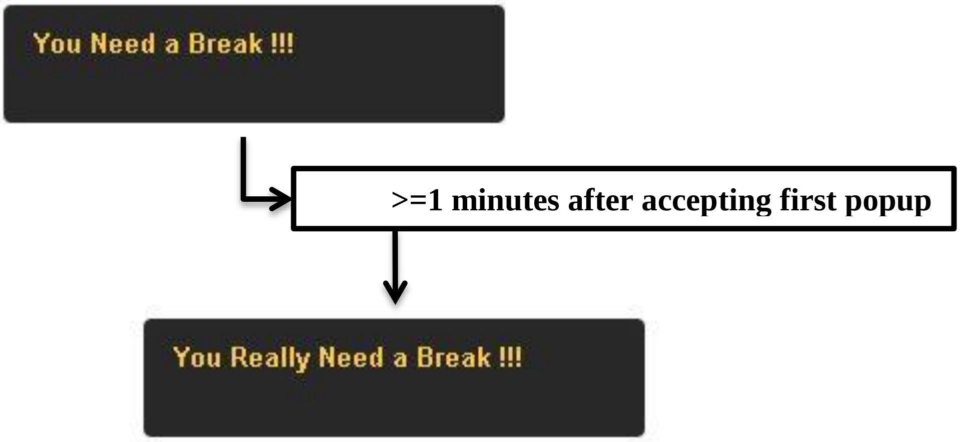

Figure 5.5 Process of break recommendations

In order to verify the third reason, if the sculpture + screen prototype are affecting the results by having two types of feedback. It was applied a Chi-Square test of 2x2 in order to understand if the screen and the sculpture prototype had any association or not with the types of breaks.

In Table 5.3 it can be seen that it's expected more "No" break recommendations in the screen prototype than the sculpture one. With these results, it appears to be an association between the break recommendations and the version of the application used.

Version App * Break		
	Break	
	No	Yes
Sculpture	25	73
Screen	81	22

Table 5.3 Number of break events for the sculpture and screen prototypes

The Pearson Chi-Square value was 56.875 with a degree of freedom of 1 (see Table 5.4) and there is a small chance of the observed difference be due to the sample.

Chi-Square Tests			
	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	56,875a	1	0,000
Continuity Correction[b]	54,763	1	0,000
N of Valid Cases	201		
a. 0 cells (0,0%) have expected count less than 5. The minimum expected count is 46,32.			
b. Computed only for a 2x2 table			

Table 5.4 Metrics for the Chi-Square test

With this sample it's more likely to have more "No" breaks recommendations than "Yes" recommendations. Having more "No" recommendations means that users might be spending less time working at the computer or they do not fell too stress and have better postures when working with these versions. These factors: stress, posture and time spend in the computer had a strong impact in the user need of a break.

Having a higher percentage of real breaks detected (251,72%) might be the reason for the user do not received more break alerts. If the users spend less time in front of the computer it's natural that the application would not achieved the necessary requirements to start recommending breaks.

5.3.6. No feedback – pre and post use

The type "No feedback" presents in Table 5.2 show the results for Pre and Post use of the application as the higher percentage of breaks recommendations with 95.24% of the total breaks events calculated. In Figure 5.3 this results are more visible and they are split into the respective categories, showing that more than 70% and 100% of the total of breaks calculated in the Pre and Post use respectively were all from break recommendations.

The "No feedback" version provided results that went against the expected, because users were already accustomed with the feedback versions and all the information provided. When it's taken all the feedback – break alerts and Log information – users complain about having missed of all that information. *P3 – "I miss the information about distance and presence, because once in a while I went look to see how my behaviour was." P4 – "I felt that the break alert was missing (...) and the Log."*

Surprisingly in one case in the Post use, the user complains about doing more breaks because of the lack of feedback. *P2 – "(...) I choose to do more breaks, because I do not have sure about for how long I was at the computer".*

5.3.7. Intuitive interaction

The intuitive interaction condition consisted of the addition of an interactive screen prototype that was used to enable users to quickly respond to the just-in time prompting mechanism. The screen prototypes are characterized by the popups show to users when they need a break and this popup do not vanish after a while. In order to make them disappear by accepting the recommendation or not it was use the screen as a way of interaction to do so.

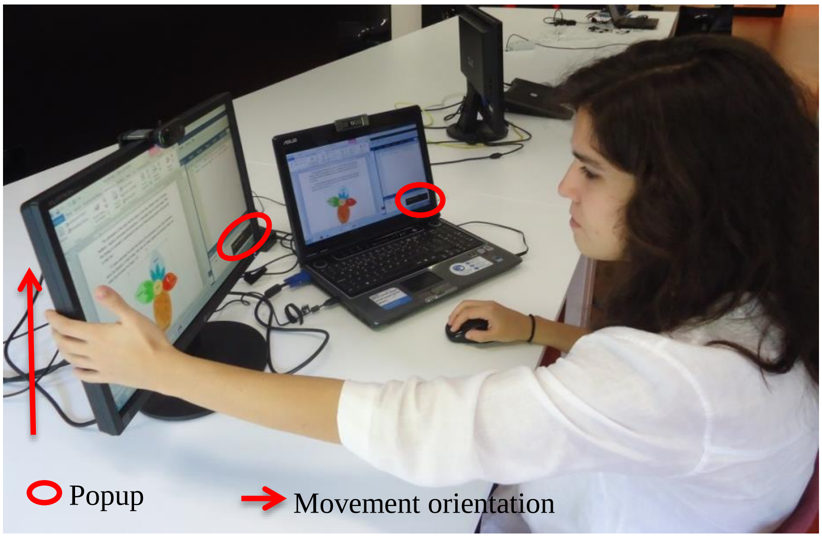

Figure 5.6 User making a delay movement when a popup recommendation show up

With the screen it was used the combination of the movements slide up and down to accept or not the break recommendation. The slide down represented an acceptance and the slide up a no/delay (see Figure 5.6). The movements performed with the down gesture were well accepted between the participants and did not cause any miss interpretation. The No/Delay movement cause a bad reaction in the participants, because it was more uncomfortable to do compare with the 'Yes' gesture. *P4 – "(…) I felt that the 'No' gesture was more uncomfortable to do." P3 – "(…) not always was easy to say 'No' (…)".*

5.3.8. Conclusion

This study had the purpose to find out if it was possible to get answer to the following questions: Can feedback influence sedentary behaviour? Can ambient feedback provide impact on users in order to accept a break recommendation? Can implicit feedback provide impact on users in order to accept a break recommendation? Can just-in time prompting provide impact on users in order to accept a break recommendation?

The results present in the upper sub sections show that users that received a break recommendation tend to accept and go to a break in 69.38% of the events shown. The average time that users need for really go to a break after receiving the alert it's from six minutes and thirty one seconds (see Figure 5.7).

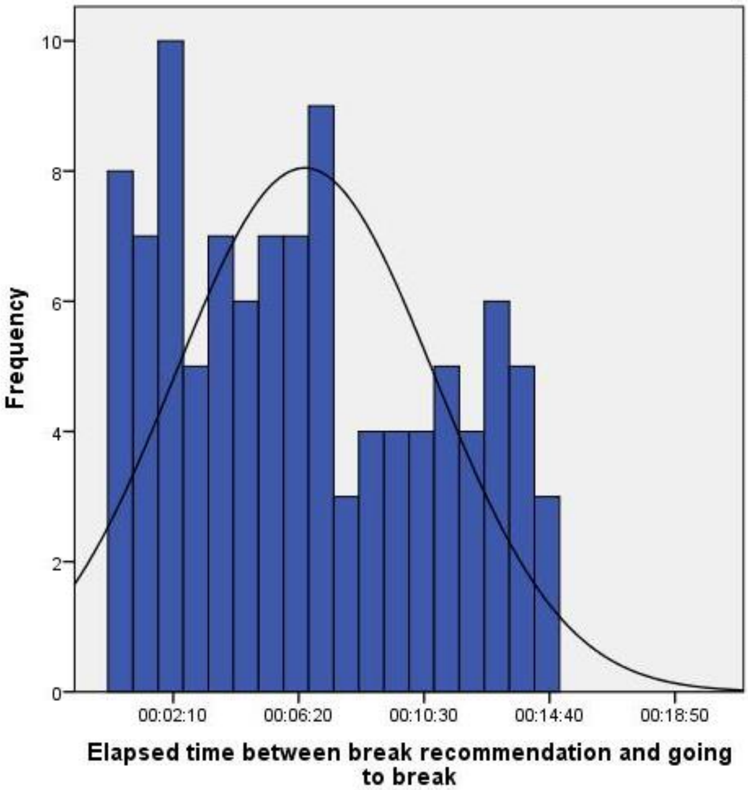

Figure 5.7 Elapsed time that users take to do a break after receiving a break recommendation

The results collected with the study show that user liked to use extra pieces (screen and sculpture) to receive output of their behaviour. Users do not feel uncomfortable by using something so visible in their surrounding environment.

Curiously the participants felt split between the three different ways of receiving feedback, all of them prefer different versions and in the end none stand up as being the favourite and easier to understand.

6. Conclusion

The goals of this thesis were to develop an application that helps people break their sedentary behaviour at work and have the application compatible in both Windows and Mac computers. The development of the application was achieved, but just for Windows computer. Unfortunately the compatibility for Mac could not be accomplished, because the frameworks/libraries to use the web camera were not fully developed for Mac computers.

Secondly, we aimed to develop a computer application that can sense when people need a break. To achieve this, we studied three different variables: posture, stress and involvement. We found these to be well predicted from the web camera as well as through the mouse and keyboard events. This study provided four contributions about the measurements that could be used to infer the predictors mentioned above.

Thirdly, we wanted to evaluate different behaviour change strategies in order to understand if different types of feedback would have some influence in the user’s acceptance or refusal of a break recommendation. To achieve this goal we implemented a second study with the findings of the first study and with the development of three different types of feedback – ambient, just-in time and intuitive.

6.1. Contributions

The contributions collected with the progress of Study I were the metrics for: posture, stress, involvement and need of a break. In this study it was possible to verify that posture can be achieved based on the distance between the user and the screen. The stress can be inferred from the movement detected in front of the computer. The user involvement can be collected through mouse activity. The need for a break could be predicted with posture and stress metrics.

The posture was related with the user viewing depth to the computer. The distance from user to the screen was collected through face detection and the squares height that identify a face in the images taken with the frontal web camera of the computer.

The stress was associated with the movement captured with the motion detector. The motion detection was performed in the pictures collected and when movement was detected it would be represented as Cartesian coordinate point. From the coordinates collected the *y*-coordinate was the one that presented a higher relation with the stress.

The involvement/engagement was related with the mouse events – left click, right click and mouse movements. From the three events associated the mouse movements and the left clicks were the ones that presented a good connection with involvement.

From the three previous measures it was deducted the need for a break through the distance from screen and the movement captured with the web camera – posture and stress predictors respectively.

The contributions collected with the Second study are related with the prototypes developed and the influence that they had in order to users accept to do a break. In this study it were developed two prototypes along with students from the IID course, the screen and the sculpture as ways to provide feedback to the users when they should do a break away from the computer.

In the screen prototype it was noticed that users were in good positions in front of the computer more often than in the other implementations. While in the sculpture prototype the movements of the cubes provided visual cues to the user but also provided knowledge to the surrounding people. The results collected with the study reveal that the users felt influence by the application to do breaks, since more than half of the break recommendations presented were accepted.

6.2. Limitations

This work suffers some limitations duo to extra factors and implementations decisions. The extra factors are related with the sample size and the length of the study from the application to help break the sedentary behaviour. Delays suffered in the previous study (Study I) caused that this application was only finished in the beginning of a holyday period. In holidays periods it's less frequent to have lots of students working at the University. Due to this constrains and the fact that the users must have a Windows computer, it was only possible to have four testers. The length of the study was choose having into consideration the holiday period and the time that users were going to spend working. After previous knowledge on the work that the possible participants would develop, it was decided that more than ten days it would be too much for them.

Particular design decision made, were not the best choices, because they had a negative impact on the users reactions. One of these choices it's present in the implicit feedback - wallpaper prototype - in the pictures chosen to represent different levels of stress. In this prototype it was presented four images (dolphin, turtle, cheat + gazelle and lion) where two of them (turtle and cheat + gazelle) get the users confused, letting them make wrong assumptions. The just-in time recommendations and the intuitive recommendation – screen and screen + sculpture prototypes – presented a problem in the movements that the user had to perform when he would like to delay the break recommendation. The gesture was made in the back of screen with a movement/touch from bottom to top. This approach was not the best since the users complained about the uncomforted produce. Since the method was not too tested, it's important to comprehend if the gesture is really "bad" or if the distance that the screen is it's producing interference.

Although I faced some problems and had made some bad assumptions, I still believe in the overall approach taken to predict the need of a break. Since it was tried to create an application that it was the less annoying and disruptive as possible to users workflow. At the same time, attempts to give users insight about their daily behaviour when working at the computer, to encourage them to be the less sedentary as possible.

The strategy follow with this system it's very different from the one proposed for the GetWitIt application. The goals establish for the GetWitIt were: to create a system that could motivate the user and other co-workers through messages to take breaks together. The user might chose the time of the day to do breaks or the frequency and the system should estimate when it's a good time and which people it should send a message. From the GetWitIt goals and the real implementation (see Appendix I - GetWitIt article) it was realize that the path to follow should be different, because the system created was not having into account the user behaviour and need to do a break.

6.3. Future Work

For future work I would like to develop further and do some improvements in the application to break sedentary behaviour. The main alterations would be made especially in the prototypes to provide feedback to users, since it was there that it was detected more problems and miss interpretations. Other important goals are: to realize the study with a higher duration and with more participants. One important requirement for the participants it's that they should not be only students, but as well administration stuff since having a diversification of users will improve the results and provide more insights.

7. References

[1] M Shields and M Tremblay, "Sedentary behaviour," *Health Reports*, vol. 19, pp. 19-30, June 2008. [Online]. http://www.statcan.gc.ca/pub/82-003-x/2008002/article/10599-eng.pdf

[2] N Jafarinaimi, J Forlizzi, A Hurst, and J Zimmerman, "Breakaway: An Ambient Display Designed to Change," in *CHI'05 extended abstracts on Human factors in computing systems*, New York, April 2005, pp. 1945-1948. [Online]. http://dl.acm.org/citation.cfm?id=1057063&bnc=1

[3] J Lin, L Mamykina, S Lindtner, G Delajoux, and H Strub, "Fish'n'Steps: Encouraging Physical Activity with an Interative Computer Game," in *UbiComp 2006: Ubiquitous Computing*, California, 2006, pp. 261-278. [Online]. http://www.ics.uci.edu/~lindtner/documents/Lin_FishnSteps2006.pdf

[4] Fitbit Inc. (2013) fitbit. [Online]. http://www.fitbit.com/uk/home

[5] Y Medynskiv, S Yarosh, and E Mynatt, "Five strategies for supporting healthy behavior change," , Vancouver, 2011, pp. 1333-1338.

[6] LR Pina, E Ramirez, and WG Griswold, "Fitbit+: A behavior-based intervention system to reduce sedentary behavior," in *2012 6th International Conference on Pervasive Computing Technologies for Healthcare*, San Diego, CA, 2012, pp. 175-178. [Online]. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6240381&tag=1.

[7] SJ Marshall and E Ramirez, "Reducing Sedentary Behavior A New Paradigm in Physical Activity Promotion," *American Journal of Lifestyle Medicine*, vol. 5, pp. 518-530, 2011. [Online]. http://ajl.sagepub.com/content/early/2011/02/10/1559827610395487.full.pdf+html

[8] CW Chen, T Määttä, KBY Wong, and H Aghajan, "A Collaborative Framework for Ergonomic Feedback Using Smart Cameras," in *2012 Sixth International Conference on Distributed Smart Cameras (ICDSC)*, 2012, pp. 1-6. [Online]. http://wsnl.stanford.edu/papers/ICDSC2012.pdf

[9] Microsoft. Windows | Dev Center - Desktop - Notifications. [Online]. http://msdn.microsoft.com/en-us/library/windows/desktop/aa511497.aspx#howoften

[10] SE Hudson et al., "Predicting Human Interruptibility with Sensors: A Wizard of Oz Feasability Study," in *roceedings of the SIGCHI conference on Human factors in computing systems*, April 2003, pp. 257-264. [Online]. http://homes.cs.washington.edu/~jfogarty/publications/chi2003.pdf

[11] J Begole, NE Matsakis, and J Tang, "Lilsys:Sensing Unavailability," in *CSCW '04 Proceedings of the 2004 ACM conference on Computer supported cooperative work*, New York, November 2004, pp. 511-514. [Online]. http://www.interruptions.net/literature/Begole-CSCW04-p511-begole.pdf

[12] M Danninger, T Kluge, and R Stiefelhagen, "MyConnector - Analysis of Context Cues to Predict Human Availability for Communication," in *Proceedings of the 8th international conference on Multimodal interfaces*, November 2006, pp. 12-19. [Online]. https://cvhci.anthropomatik.kit.edu/~stiefel/papers/MyConnector_ICMI06_final.pdf

[13] European Agency for Safety and Health at Work. (1998-2013) European Agency for Safety and Health at Work - Stress - definition and symptoms. [Online]. https://osha.europa.eu/en/topics/stress/definitions_and_causes

[14] Oxford University Press. (2013) Oxford Dictionaries - Definition of fidget. [Online]. http://oxforddictionaries.com/definition/english/fidget

[15] Dhanika. (2013) Consumer Instinct - Did You Know? Fidgeting can be good for you. [Online]. http://www.consumerinstinct.com/foodhealth/did-you-know-fidgeting-can-be-good-for-you/

[16] Oxford University Press. Oxford Learner's Dictionary - Definition of involvement noun. [Online]. http://oald8.oxfordlearnersdictionaries.com/dictionary/involvement

[17] V Navalpakkam and E Churchill, "Mouse Tracking: Measuring and Predicting Users' Experience of Web-based Content," in *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, May 2012, pp. 2963-2972. [Online]. http://dl.acm.org/citation.cfm?id=2208705

[18] C Epp, M Lippold, and RL Mandryk, "Identifying Emotional States using Keystroke Dynamics," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Vancouver, Canada, May 2011, pp. 715-724. [Online]. http://hci.usask.ca/uploads/203-p715-epp.pdf

[19] Farlex, Inc. The Free Dictionary by Farlex - Posture. [Online]. http://www.thefreedictionary.com/posture

[20] C Hidalgo. (2013) No end insight - How to Straigther up, with Posture Exercices. [Online]. http://noend-insight.com/2013/02/how-to-straighten-up-with-posture-exercises/

[21] WorkSafeNB. (2010) Office ergonomics - Guidelines for Preventing Musculoskeletal Injuries. [Online]. http://www.worksafenb.ca/docs/OFFICEEdist.pdf

[22] Openshaw,S. Allsteel;Taylor,E. Allsteel. (2006) Ergonomics and Design a Reference Guide. [Online]. http://cms.allsteeloffice.com/SynergyDocuments/ErgonomicsAndDesignReferenceGuide

WhitePaper.pdf

[23] A Davison. (2013) NUI Chapter 7. Face Detection and Tracking. [Online]. http://fivedots.coe.psu.ac.th/~ad/jg/nui07/index.html

[24] NUI Chapter 3. Motion Detection. [Online]. http://fivedots.coe.psu.ac.th/~ad/jg/nui03/

[25] Kristian Kraljic, kra.lc. KRISTIAN KRALJIĆ - Java – Global (low level) Keyboard / Mouse Hook. [Online]. http://kra.lc/blog/2011/07/java-global-system-hook/

[26] S Siegel and N Castellan,.: McGraw-Hill Book Company, 1988, ch. 9.3-9.5, pp. 235-256.

[27] G. Cunha, M. Martins, S. Sousa, and F. Oliveira,.: LIDEL - Edições Técnias, Lda, 2007, ch. 4.2.2, pp. 116-120.

[28] Arduino SA. Arduino - Capacity sensor by Paul Badger. [Online]. http://playground.arduino.cc//Main/CapacitiveSensor?from=Main.CapSense

[29] Blue Point Engineering. (2007) Servomotor Information. [Online]. http://www.bpesolutions.com/bpemanuals/Servo.Info.pdf

[30] (2012, April) Laurid Meyer.com - Electronics, Arduino, Java | Java Serial single Byte/8Bit Communication. [Online]. http://www.lauridmeyer.com/2012/05/arduino-java-usb-serial-communication-all-you-have-to-know/javaserialsinglebytecom/

[31] Heinrich Spreiter SwingFx.ch. (2011) SwingFX - Twinkle. [Online]. http://www.swingfx.ch/twinkle

[32] Stack Overflow - Can I change my Windows desktop wallpaper programmatically in Java/Groovy? - Stack Overflow. [Online]. http://stackoverflow.com/questions/4750372/can-i-change-my-windows-desktop-wallpaper-programmatically-in-java-groovy

[33] WallpapersBuzz - crop Dolphin swimming wallpaper. [Online]. http://www.wallpapersbuzz.com/reviews/cropimage/3039

[34] Kauai Real Estate Blog - Tired Turtle in Poipu…no, not a realtor. [Online]. http://susannakunkel.wordpress.com/2008/11/08/tired-turtle-in-poipu-no-not-a-realtor/

[35] Cheetah chasing fawn - Andy Rouse. [Online]. http://500px.com/photo/1010307

[36] Fanpop! - All About Lions - the male lion. [Online]. http://www.fanpop.com/clubs/all-about-lions/images/7875263/title/male-lion-photo

[37] E Karapanos, M Hassenzahl, and J Martens, "User experience over time," in *CHI 2008 Proceedings*, Florence, Italy, April 2008, pp. 3961-3964. [Online]. http://ekarapanos.com/UX_over_time.pdf

[38] opencv dev team. (2011-2013) OpenCV 2.4.6.0 documentation - Welcome to opencv documentation! [Online]. http://docs.opencv.org/

[39] MathsIsFun.com. (2012) MATHisFUN - Polar and Cartesian Coordinates. [Online]. http://www.mathsisfun.com/polar-cartesian-coordinates.html

[40] Youtube - Moving Cubes (no tape). [Online]. http://www.youtube.com/watch?v=LuxMeccVrCY

8. Appendices

Appendix I - GetWitIt article

GetWitIt: Breaking sedentary behavior through micro-meetups

Kara Rennert
Carnegie Mellon University
M-ITI University of Madeira
Pittsburgh, PA
Kara.cmu@gmail.com

Ana Neto
University of Madeira
Funchal, Madeira
ana.b.neto@gmail.com

Maria José Ferreira
University of Madeira
Funchal, Madeira
maria_joserf@hotmail.com

Micaela Sousa
University of Madeira
Funchal, Madeira
micas.s.s.sousa@gmail.com

CHI'13, April 27 – May 2, 2013, Paris, France.
ACM 978-1-XXXX-XXXX-X/XX/XX.

Abstract

There are now many devices and applications to track physical activity, in order to support self-awareness and encourage healthy behavior. These devices vary in how they make users aware they have been sedentary too long, and how they persuade them to take a break and be physically active. None, however, make a persuasive suggestion for how the individual could be physically active.

In this paper we describe the GetWitIt desktop application, designed to use crowdsourcing and the influence of social networks to encourage students to take both spontaneous and planned breaks to curb the effects of sedentary behavior.

We expect that our micro-meetup approach to encourage taking breaks will decrease sedentary behavior.

Author Keywords

Sedentary behavior, micro-meetups, social influence, health behavior change, personal health management.

ACM Classification Keywords

H.5.2 [Information Interfaces and Presentation]: User Interfaces – prototyping, theory and methods, user-centered design.

General Terms

Design, Human Factors, Theory.

Introduction

Research about the ill effects of sedentary behavior is now well known throughout the health community and is becoming more widely known in the general population. Research suggests that sitting for long periods of time, independent of overall physical activity level, has a negative impact on both short-term and long-term health. During working hours people tend to sit between 4.5 to 9 hours [1, 2, 3]. Some of the consequences are increased risk of weight gain, metabolic disruption, and premature mortality [1, 4]. There is also an increasing body of research suggesting that breaking up sedentary periods with short periods of physical activity has a positive effect on health [1, 5]. This means that these negative consequences could be mitigated even with small changes in behavior.

Extensive psychological research has been done in the area of influencing behavior change. This research is the foundation for persuasive technology. Several persuasive technology projects have attempted to encourage people to become more physically active when they detect sedentary behavior.

Breakaway is a sculpture that sits at the user's desk. It continues to slump over as the user remains sedentary, and sits upright and appears "healthy" when the user takes regular breaks. In this way the user is reminded when to take breaks through a visual depiction of the ill effects of sitting for too long. Results showed that the user appreciated being able to ignore Breakaway when busy, and that the user took more breaks while using it [11, 16].

Fish 'n' Steps encourages users to take more steps each day. The user's step count is linked to the emotional state, growth, and activity of a virtual fish. This fish is displayed on a kiosk in a common area of the user's workplace in a virtual fish tank with the fishes of other users [11, 17]. Similar to Breakaway, there is a visual depiction of the sedentary status of the user. In this instance, it is translated to overall health of a fish.

FitBit+ is a system that uses its automated step logging to detect sedentary behavior and then prompt users to take a walking break[1]. FitBit alleviates the burden of self-reporting by automatically tracking a person's daily step count, but it cannot intervene when undesired behavior is occurring. The design opportunity explored by FitBit+ is to create an *in-situ* behavior change intervention by extending this self-tracking device to send messages to users about when it is time to get up and get active.

Breakaway, Fish 'n' Steps, and FitBit+ all solve the problem of telling the user when it is time to take a break, but they do not tell the user what to do. How should the user take a break? GetWitIt seeks to solve that problem by using social intelligence to suggest break time activities, and the influence of social networks to encourage users to take these breaks. We will also use it to test the persuasive power of using novel and frequently changing acceptance messages to try and motivate users to accept these invitations to take breaks.

How to effectively use social networks to support everyday behavior change represents an interesting challenge. [6] Extensive research shows that attitude change is greatest following public behaviors. [12] Public commitment theory argues that having an audience for behaviors tends to magnify the effect that behavior has on attitudes [9, 12]. Digital technology provides an efficient way to create an audience.

Some research has addressed the social aspects of physical activity, finding that sharing information about activity and exercising together can increase interest, enjoyment, and motivate some individuals to increase activity [5, 8, 20, 25]. Also, when people receive tailored information that is personally relevant, it is more likely to stimulate change, adding to people's self-efficacy and outcome expectancy [23, 24]. It is evident that, although not determined by social factors, intrinsic motivation is affected by wider social interaction [10]. This research suggests that group support is key to changing behavior.

Based on this research, our application tests whether users will respond more to public events or those involving a close circle of friends, and whether they are more likely to create a public or private event. Also, are users more likely to attend an event if they know the creator?

Another important contribution of the GetWitIt application is the power of persuasion. How can we encourage users to accept an invitation to take a break? Would a novel and positive message work? In other words, will a novelty in design motivate users to take more breaks?

Breakaway and FitBit+ also raise the issue of the significance of interruption in technology. The research done by Gillie *et. al* [6] focuses on the effects of different types of interruptions on computer-based tasks. Her work demonstrates that if the interruption is not a heavy cognitive load, regardless of its length, it does not disrupt the performance of the main task. It is also important that messages are received *in-situ*, that is delivered to where the user is working. [1] Breakaway worked well because it was highly visible in the users workspace, but did not disrupt the user from their main task.

The GetWitIt invitation notifications have been designed to gain user attention, but minimize user distraction. Notifications appear for a short duration and disappear. The goal is to provide the message *in-situ*, but to minimize interruption. The message is also brief with a minimal response required to reduce cognitive load. If the user chooses to answer right away, it should take minimal thought and effort.

Based on the theories discussed and questions raised, we have designed a system to encourage university students at the University of Madeira (UMa) to take more breaks during the day.

We intend to leverage the influence and enjoyment factors of social networks through this application to encourage people to increase their daily break time activity. We believe that by "crowdsourcing" the suggestions for activities from within the community, the activities will be more personally relevant for the user, and they will be more likely to participate and maintain their participation over time. We are also employing a novel RSVP design to encourage users to accept invitations and break their sedentary behavior.

System Description

Students, researchers, faculty and staff at universities all spend a majority of their day sitting behind a computer. Our research showed that students at UMa spend up to 9 hours per day at their computer on the University campus, with an average of over 5 hours.

The number of breaks changed according to their workload throughout the semester, varying between 2 to 4 breaks per day. The most common duration for breaks is 10 to 15 minutes with only 12% of breaks over 15 minutes.

Breaks are a social activity for students: 40% specifically take breaks because they want to meet with friends, and 95% prefer not to take their breaks alone.

The number of activities that students could think of for breaks was low, with only 9 different activities suggested throughout our user research and 50% of the students limiting all their break activities to getting something to drink or eat.

65% of the students were willing to try an application to reduce their sedentary behavior. When asked about what characteristics they would prefer in the application, many mentioned not wanting it to nag them, but rather be compelling, fun and social.

The GetWitIt application is a simple application to encourage people to take more breaks throughout the day by arranging micro-meetups. The application runs on Windows computers, and can be found in the system tray whenever it is minimized.

The strategy for using micro-meetups is founded on 5 proven strategies currently being used in effective health self-management interventions. [8] We have adopted them to support behavior change in the GetWitIt application:

- The events refer to activities that are
    - specific,
    - short-term and
    - actionable.
- It is expected that users would they choose activities they are confident they can attain.
- Users receive cues-to-action through reminder messaging and the system design allows users to experiment, seeing how they could integrate it into their daily life.

Our design also follows design strategies for persuasive technologies. These are strategies for motivating behavior change in everyday life through technology that specifically accounts for the intersection of the technology with the user’s complex social world. [11]

We kept 3 of the 8 design strategies in mind when creating the interaction design for our application. In particular, we focused on making sure the system was:

1.) **Unobtrusive**
2.) **Public**
3.) **Trending/Historical**

First, **Unobtrusive**. Defined as presenting and collecting data that does not interrupt everyday life or call attention to the user. We aimed to also support ignorability as suggested by Cognitive Dissonance Theory [14]. The GetWitIt design supports this strategy by using non-disruptive meet-up messages, RSVP’s, and event rating windows without interrupting the users' workflow, regardless if he accepts the invitation to take a break or not.

Second, **Public**. The technology should not make the user uncomfortable if seen by the general public. The GetWitIt application is a windows desktop application that does not have any features that should make a user uncomfortable if seen by the general public.

Third, **Trending/Historical**. The technology should provide reasonable and accessible information about the user’s past behavior as it relates to their goals. In order to meet this design strategy, users will be able to see if they have met their daily, weekly, and monthly goals. They will also be able to see all activities they were invited to over a period of time and which ones they agreed to participate in. This way they can track their improvement, or see when their activity may have trailed off.

Here we see how users can create their specific, short-term, and actionable event and invite other users to it (see Figure 8.1). The user can create any type of event they want. An event simply has a title, a description and the date and time it will occur. The event creator can also specify the time the notification for the event will appear on the invited users' desktop, reminding them to go.

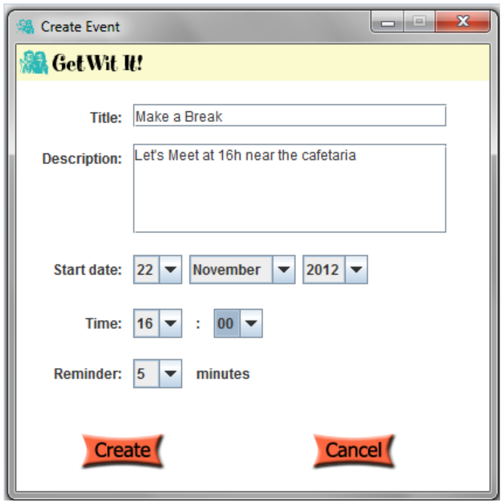

Figure 8.1 – GetWitIt – Create Event – first step

Invitations can be sent to everyone (public events), all the user's friends or specific people (Figure 8.2). The user can also create a friend list using the Profile window. This makes inviting all friends to an event much easier.

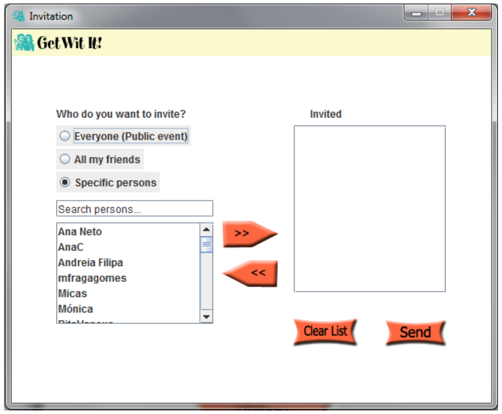

Figure 8.2 – GetWitIt – Create Event – second step

Users can check upcoming events they have been invited to in the Event List (Figure 8.3). The user can also see past events by clicking the Old Events button.

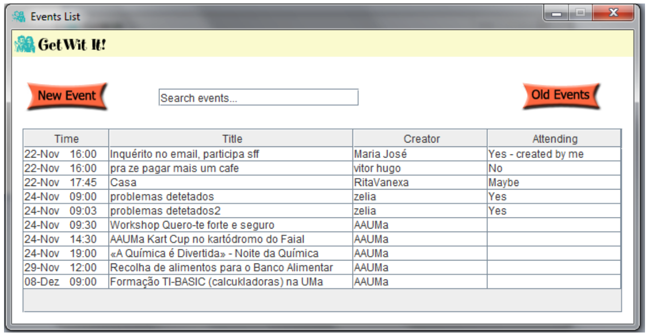

Figure 8.3 – GetWitIt – Event List

Invited users will be able to see the number of people that have accepted the invitation (Figure 8.4).

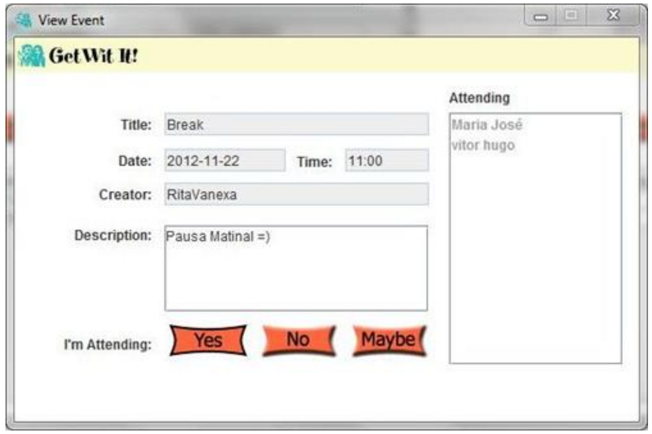

Figure 8.4 – GetWitIt – View Event

We believed that knowing not only the creator but also other people intending the event would be important to users.

A notification appears on the bottom left corner of the user's desktop when the time for the event is approaching, the amount of notice previously chosen by the event creator (Figure 8.5).

Figure 8.5 – GetWitIt – Notification 3 button version

Figure 8.6 – GetWitIt – Notification 1 button version

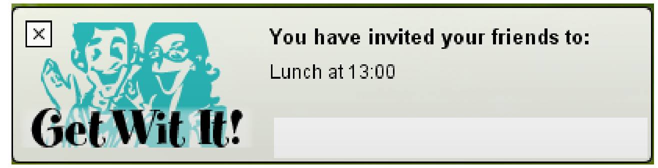

Figure 8.7 – GetWitIt – Notification for the event creator

Half of our users would be able to RSVP to the event using the yes/no/maybe notification buttons. If the user doesn't react in 10 seconds, the notification fades out automatically. This was designed to keep interruption to a minimum, while maximizing awareness. Users would see the notification while working, but could also ignore it easily.

The other half of our users would only have one button to RSVP to events (Figure 8.6). This "yes" button would rotate between "Yes", "Sure!", "Yeah!", and "Great!". We wanted to see if users would be more likely to accept the invitation if there was only this option, or canceling or ignoring the invitation. We believed that positive and humorous alternatives to the standard "yes" would also persuade users to accept invitations.

After an event was created, a reminder notification would be received by the creator as the meeting time was approaching (Figure 8.7).This notification would also appear for 10 seconds and then fade away, in order to not be intrusive.

Those that attended an event would be asked to rate their enjoyment of the activity after participating (Figure 8.8).

Figure 8.8 – GetWitIt – Rate Event

We assume the user went to all the events they rated. Based on that information, the application shows the number of events the user attended on that day, week and month (Figure 8.9). This screen provides reasonable and accessible information for the user about their past behavior. This allows users to compare it to their goals.

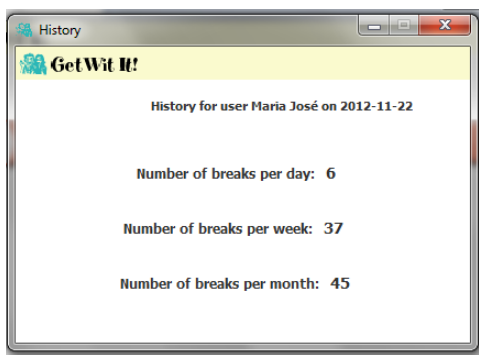

Figure 8.9 – GetWitIt – History

In "The Paradox of Choice", Barry Schwartz [21] discusses how paralysis occurs when faced with too much choice. When making a decision is too hard, people don't make a choice. And even if they do make a choice, they are less satisfied than if they had fewer choices. Why does this happen? Schwartz states that as options increase, so do expectations. So increasing choices, increases expectations and paralysis, and decreases satisfaction. How can we limit choice just enough to maximize choice and satisfaction?

Malcolm Gladwell talks about horizontal segmentation in "Choice, Happiness, and Spaghetti Sauce," [22] basically stating there is a disconnect between what people say they want, and what they actually want. Many may say they want the same thing, but reality shows that different people want different things. In order to meet that need, there needs to choice.

Sheena Iyengar in "The Art of Choosing" [23] discusses how culture impacts choosing and found that it is a mistake to assume that everyone thrives under the pressure of choosing alone. Some people feel just as satisfied with a choice if someone they respect chooses for them.

We were also curious how feelings about choice would have an impact. Would users feelings about choice affect their experience of the GetWitIt application? Would limiting the RSVP to only a yes, instead of yes/no/maybe encourage users to accept versus ignoring the message? Would it have any effect on the attendance?

Study Proposal

Method

For our study, we recruited 25 participants from UMa that currently own and use a PC computer. Users were aged from 21 to 48 years old (average 25 years old, standard deviation 6,6) and evenly divided in gender (12 males, 13 females). All users are currently university students.

The participants used the GetWitIt application for 2 weeks.

The GetWitIt system keeps track of the number of breaks the user takes based on the number of activities the user rates. If they rate an event, it is assumed that the user attended the event. The GetWitIt system also tracks the number of events the user attended to which he had accepted invitations.

We interviewed study participants at the beginning of the testing period to find out how many breaks they typically take during the day, what types of activities they normally take a break for, and ideally how often would they prefer to take breaks.

We implemented an A/B testing scenario into our prototype. Half the users had an interface with YES/NO/MAYBE options for responding to an invitation, while the other half received only a YES option. This "yes" option also varied randomly with other encouraging messages, such as "Great!", "Sure!", or "Yeah!". Half the time, the message is neutral (Yes) and half the time the application shows a positive message instead ("Great!", "Sure!", or "Yeah!").

We also tested to see if there are more yes responses in the B version of the notification, as well what version produces a higher rate of attendance from those yes responses to invitations.

During the testing period we did an online questionnaire to the users using Likert-type scales. We wanted to know about the GetWitIt application usability and how the users felt about the notifications, namely how disruptive they were for them.

We also conducted user interviews to selected users during the testing period to get an insight on how users reacted to the notifications and how they actually used the application during the testing period.

At the conclusion of the two week trial period, users were interviewed again to find out if they felt the application had encouraged them to take more breaks than they had previously and what they felt about the interface and its ease of use.

Results

Our study had 30 total participants, 5 of which consisted of our research team. The research team created some events in the first two days of the study, as shown in the graph below (Figure 8.10. in green). These events were created in order to encourage participation, as we believe it is easier to figure out how the application works with some events already created. In total, 226 events were created over the 2 week study.

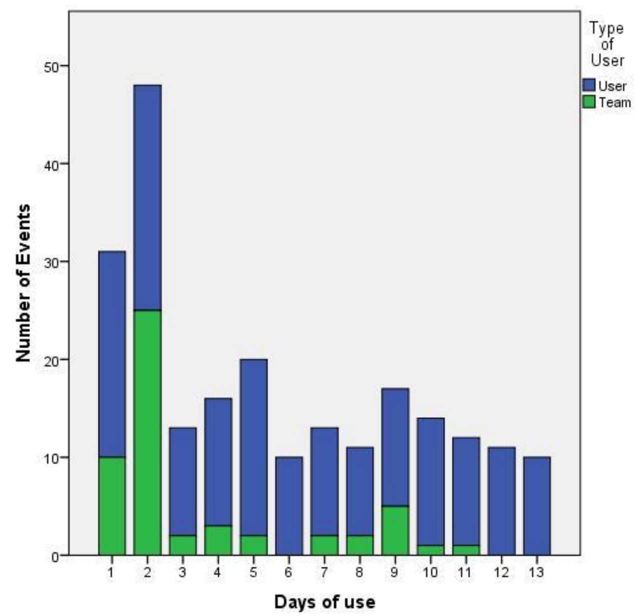

Figure 8.10 – Number of events per day

There was some novelty effect on the first 2 days of the study, in which 31 and 48 events were created, but then usage stabilized with an average of 14 events per day and 5.35 invitations per event.

Users tended to create events that replicated their normal breaks both in terms of people invited and type of activity. Creation of public events was uncommon (only one was created by a user outside the research team). Users stayed within their social networks, inviting either specific friends, or all their friends from their friends' list.

Category	Freq. (%)	Mean (Day)	SD
Coffee and Snack	75 (33%)	6	4
Go outside	20 (9%)	6	3
Lunch	20 (9%)	7	4
Message	25 (11%)	4	3
Non specified break	22 (10%)	6	4
Other	13 (6%)	7	4
Reminder	11 (5%)	6	3
Social	9 (4%)	6	4
Test	31 (14%)	2	0
Total	226 (100%)	-	-

Table 8.1 - Categories of events frequency

They also maintained the types of activities they had before using the application (see Table 8.1). Activities related to eating or drinking still were the most popular, accounting for over 40% of the use cases.

In general, users favored specific event categories (Figure 8.11). On average each user created events in 3 different categories (with standard deviation of 1.9).

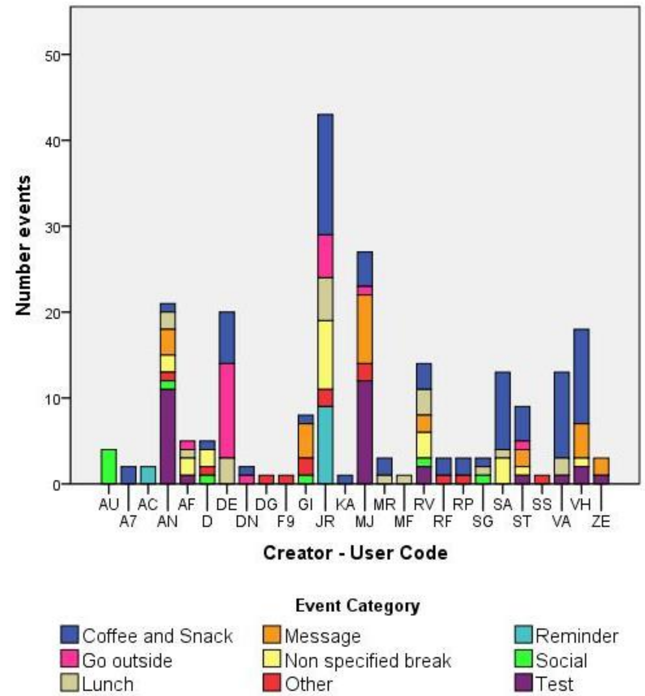

Figure 8.11 - Number of events per creator with detailed category information

There were some surprising use cases, however. Users tried to use the application to send each other messages or reminders (Category "Message" or "Reminder" respectively). This accounted for a significant number of use cases with over 15% of the use cases.

For instance, 8 events were created by users to remind each other of buying lunch tickets. Lunch tickets at the university are cheaper if bought some time ahead, so the users created this type of event to make sure they got the cheaper price.

The users created "Message" type events for 2 different purposes:

- Send event related messages (for instance, to ask the event creator about location details).
- Send other type of messages (for instance, a user created a message asking another if she wanted a lift home).

Regarding the A/B testing done with the two versions of the application, we found that in both cases the number of invitations with no response was very high, 73% for the 1-button version and 75% for the 3-buttons version. This might result from the user not being logged in the application at the time the notification should appear, the user ignoring the notification or the user closing the notification using the close button.

The high number of no responses made it impossible to do statistically relevant analysis of the data regarding the actual responses.

We were expecting a higher number of acceptances for events when the message in the notification buttons was a positive message. This was not the case, as with both versions of the application (see Table 8.2), show a slightly higher acceptance rate for the neutral messages.

	Yes	**No or Maybe**
Positive	52 (83%)	11 (17%)
Neutral	70 (84%)	13 (16%)

Table 8.2 – Variation on attendance by positive/neutral button messages

In the interviews some users referred not being aware of different messages in the notification buttons.

MR - *"(...) I think I only had one button but I'm not sure, (...) I don't remember if the buttons had different messages (...)."*

DE – "(…) *The buttons always had the same messages in my version (…)."*

The slightly higher number of acceptances with neutral messages might result from users choosing to respond to an invitation using the View Event window (in which the message was always neutral as seen in Figure 8.4) instead of the notification.

RV - *"(…) I used the event list just because I was curious to see who has invited me (…) and if anyone that I know has invited me, I accept it right away (...)."*

SA – "*(…) I used to go to the events' list to see what events I might have been notified about without noticing (…) I'm not always looking at the computer screen and I might not have seen some event. (...) if there is an event that I would like to go I use it to see who is going too (...)"*

Regarding the number of buttons in the notification (see Table 8.3), the data was, again, not statistically relevant.

	Yes response	**No response**
1 button (Yes)	74 (21%)	256 (79%)
3 button (Yes No Maybe)	128 (20%)	526 (80%)

Table 8.3 - Variation on attendance depending on the notification button

However, the qualitative data from the interviews made us realize that users preferred the version with 3 buttons, as they felt the application was forcing them to choose "Yes" when they did not want to.

ST – "*(...) Unfortunately (...) I will say that I don't like the buttons, because I only have one, (...) next time you give me an application give me three buttons! (...)."*

We believe users felt more comfortable and even more motivated to take breaks because of the social intelligence behind the GetWitIt application: they were invited by someone they know (friends / acquaintances) instead of feeling they were being told what to do by some application.

VA – *"(…) I was worried the application might be nagging but it is only a new way of having friends inviting me for a break (…)."*

As long as the messages are from a friend or acquaintance, users don't even seem to mind much about the message itself. Even if they receive the same message several times it doesn't seem to have a negative impact on their answer or their experience with the application.

SA – *"(…) Most my events are coffee breaks, I just write «Coffee» and they come as long as they can take a break at that time (…)."*

RV – *"(…) I had never noticed that is all you write on the event message… but, yes;,I know where we usually meet, so it is enough! (…)."*

Finally the interviews confirmed that the strength of the GetWitIt application was the notification system.

Users felt more compelled to actually do a break because they had been notified ahead of time.

DE - *"(...) if I invited DN personally he always said no, but if I used the application, he would go straight away because he had been notified some time before (...)."*

DN - *"(...) it depends on what I'm doing (...) after receiving the notification I'm no longer surprised when DE comes for us to go (...)."*

They also felt it easier to realize how long they had been working without doing a break. The notifications made them achieve this higher level of awareness.

ST - *"(…) It helps to realize how much time as gone by (…) when we are working, we could lose a sense of time, when the notifications appears you see the hours and realize that you can do a break at that time(…)."*

RV -*" (…) I needed the notification to remind me to go to the event, because a person forgets these things, when one is distracted or focused on something (…)."*

This higher level of awareness was achieved without too much disruption of the users' work tasks.

JR- *"(…) it doesn't interrupt me, it draws attention but not in an intrusive way (…)."*

AN – *"(…) if the notification popped up when I was working, I knew I could ignore it and keep working because it would go away on its own after a little while (…)."*

Users seem to already be used to notifications that pop-up, so they are now accustomed to keep working despite of them.

VH – *"(…) It is like the notifications from Dropbox, messenger or the antivirus, I take a quick look but I can keep working (…)."*

Discussion and Conclusion

Our findings suggest that using a notification system on the desktop can raise the level of awareness of users without significant disruption of their work.

In line with our expectations and the work of Adamczyk *et al.* [7], we found that having the notification for the

GetWitIt application show in the corner of the desktop, with minimal information about the event (therefore little cognitive load) and fading away automatically after 10 seconds, made it minimally disruptive.

Different from most research that addresses ways to break sedentary behaviour ([1], [11], [17]), our study addressed using social intelligence to that end. Instead of having the system prompt the user with standard messages, the user was prompted by other users to have a break. This made each notification different and socially relevant.

We found that this made users more aware of the need for a break, by not losing track of time, while not feeling the application was nagging or forcing them in any way.

We believe this worked best for the users that had the friends they usually go for breaks with also using the application. Since users tried to replicate their normal breaks, the best results were achieved when the persons they wanted to invite for a break were also using the application.

Future research could explore if having an easy way to invite your friends to use the application or a way of integrating it with social networks, namely Facebook, would have an impact on the number of events created and attended by users.

We also found that users used the GetWitIt application in unexpected ways.

First, some users use the application to plan all their breaks for the day (or the next day).

Secondly, some users tried using the application for communicating with each other, creating an event just so as to make their message pop-up in the other users' desktop at a given time. Sometimes these messages where related to other events, but other were just general attempts of communicating with the other user.

With the GetWitIt application we also tried to evaluate whether minimizing the number of choices in the notification or introducing simple positive messages in the notification buttons would make it more likely that the user would accept an invitation.

The question we raised was if these changes in the notification design could increase the number of breaks the user did.

Our findings were not statistically relevant as most of the time users did not respond to the invitations they got (75% of invitations are not answered).

Future research should attempt to study whether this is due the user not being at their computer, not having the application open or because the notification was ignored. We believe that the first two cases would account for a significant number of non-responses.

Following from suggestions from the users in our study we also believe that in the future it would be interesting to include in the GetWitIt application or other similar applications:

- Sound as an optional feature in the notifications
- Chat option, as the users feel the need to communicate as a result of the social nature of the application
- Mobile version of the application, to make creating an event or viewing the event list available to the user even when he is not at the desktop.

Concluding, our study aimed at using social intelligence and a different design to reduce the sedentary behavior of users.
While our study didn't prove the usefulness of some of our design options, such as reducing the number of buttons in the notification and using simple positive messages in the button, it provided promising results regarding the awareness and disruption levels.

Acknowledgements

We thank all the volunteers, and all those that provided helpful comments.

References

[1] Fitbit+: A behavior-based intervention system to reduce sedentary behavior, http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6240381&tag=1.

[2] W. 1. Brown, Y. D. Miller, and R. Miller, "Sitting time and work patterns as indicators of overweight and obesity in Australian adults." International journal of obesity and related metabolic disorders journal of the International Association for the Study of Obesity, vol. 27, no. 1 1, pp. 1340-6, 2003. [Online]. Available: http://www.ncbLnlm.nih.gov/pubmed/ 14574344

[3] K. I. Proper, E. Cerin, W. J. Brown, and N. Owen, "Sitting time and socio-economic differences in overweight and obesity." International journal of obesity 2005, vol. 3 1, no. 1, pp. 169-176, 2007. [Online). Available: http://www.ncbi.nlm.nih.gov/pubmed/ 16652126

[4] S. J. Marshall and E. Ramirez, "Reducing sedentary behavior : A new paradigm in physical activity promotion," Building, vol. In Press, no. x, 2011. [Online). Available: http://ajl.sagepub.comlcgi/doi/1O. 1177/1559827610395487

[5] G. N. Healy, C. E. Matthews, D. W. Dunstan, E. A. H. Winkler, and N. Owen, "Sedentary time and cardio-metabolic biomarkers in us adults: Nhanes 2003-06." European Heart Journal, vol. 32, no. 5, pp. 590-597, 2011. [Online]. Available: http://www.ncbLnlm.nih.gov/pubmed/21224291

[6] T. Gillie and D. Broadbent, "What makes interruptions disruptive? A study of length, similarity, and complexity," Psychological Research, vol. 50, no. 4, pp. 243-250, 1989.

[7] P. Adamczyk and B. Bailey, "If not now, when?: the effects of interruption at different moments within task execution," in Proceedings of the SIGCHI conference on Human factors in computing systems. ACM, 2004, pp. 271-278.

[8] Yevgeniy Medynskiy, Svetlana Yarosh, and Elizabeth Mynatt. 2011. Five strategies for supporting healthy behavior change. In Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems (CHI EA '11). ACM, New York, NY, USA, 1333-1338.

[9] DOI=10.1145/1979742.1979770 [Online]. Available: http://doi.acm.org/10.1145/1979742.1979770

[10] Amy L. Gonzales, JP Pollak, Daniela Retelny, Eric P.S. Baumer, Geri Gay, 2011. A Mobile Application for Improving Health Attitudes: Being Social Matters. In CHI 2011, May 7-12, 2011, Vancouver, BC, Canada. ACM 978-1-4503-0268-5/11/05.

[11] Maitland, J., Sherwood, S., Barkhuus, L., Anderson, I., Hall, M., Brown, B., Chalmers, M., & Muller, H., "Increasing the Awareness of Daily Activity Levels with Pervasive Computing," Proceedings of Pervasive Health '06 , Innsbruck, Austria, (Nov/Dec 2006).

[12] Consolvo, S., McDonald, D.W., and Landay, J.A. Theory-driven design strategies for technologies that support behavior change in everyday life. Proceedings of CHI 2009. ACM Press (2009), 405-414.

[13] Schienker, B. R. et al. (1994). The impact of self-presentations on self-appraisals and behavior: The power of public commitment. Pers. & SocPsy, 20, 20–33.

[14] Goffman, E. The Presentation of Self in Everyday Life. New York, NY, USA: Doubleday Anchor, (1959).

[15] Festinger, L. A Theory of Cognitive Dissonance . Stanford, CA: Stanford University Press, (1957).

[16] Fogg, B.J. Persuasive Technology: Using Computers to Change What We Think and Do , San Francisco, CA, USA: Morgan Kaufmann Publishers, (2003).

[17] Jafarinaimi, N., Forlizzi, J., Hurst, A., Zimmerman, J. "Breakaway: an ambient display designed to change human behavior," In CHI '05 Extended Abstracts , Portland, OR, USA, (Apr 2005), 1945-8.

[18] Lin, J.J., Mamykina, L., Lindtner, S., Delajoux, G., Strub, H.B. "Fish'n'Steps: Encouraging Physical Activity with an Interactive Computer Game," Proceedings of UbiComp'06 , Orange County, CA, USA, (Sep 2006), 261-78.

[19] Consolvo, S., Everitt, K., Smith, I., Landay, J.A. "Design Requirements for Technologies that Encourage Physical Activity," Proceedings of CHI 2006 , Montreal, Canada, (Apr 2006), 457-66.

[20] Klemmer, R.S., Thomsen, M., Phelps-Goodman, E., Lee, R. and Landay, J.A. Where do web sites come from? Capturing and interacting with design history. In Proc. CHI 2002, ACM Press (2002), 1-8.

[21] Schwartz, B. The paradox of choice: Why more is less. New York: Ecco, (2004).

[22] Novinson, E. "Malcolm Gladwell: Choice, happiness and spaghetti sauce."

[23] Iyengar, S. The art of choosing. Twelve, (2010).

Appendix II - Entity–Relationship Model for GetWitIt application

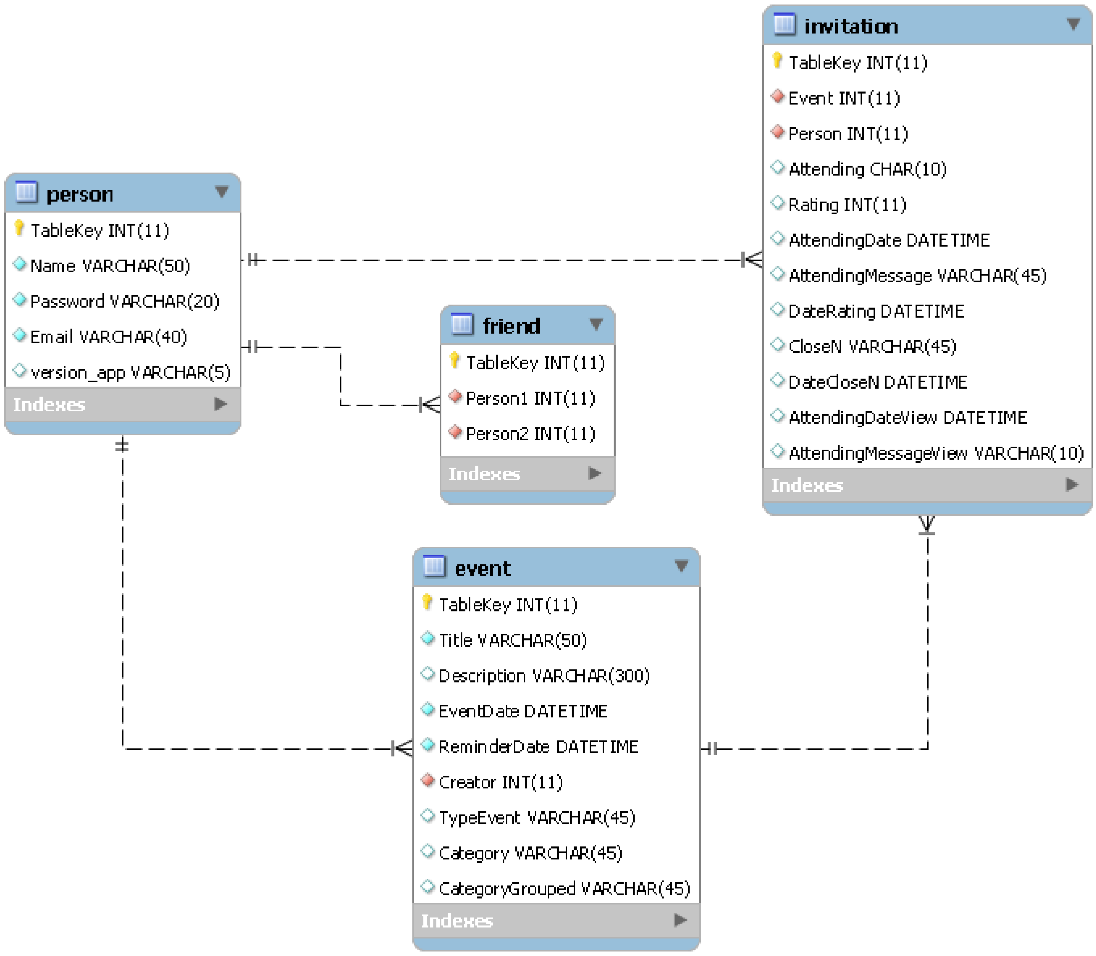

Figure 8.12 Entity-Relationship Model for GetwitIt application

Appendix III - GetWitIt Online Questionnaire

I think the GetWitIt notifications are:

- Strongly Unpredictable
- Moderatly Unpredictable
- Slightly Unpredictable
- Neutral
- Slightly Predictable
- Moderatly Predictable
- Strongly Predictable
- Don't know

I think the GetWitIt notifications are:

- Strongly Confusing
- Moderatly Confusing
- Slightly Confusing
- Neutral
- Slightly Clear
- Moderatly Clear
- Strongly Clear
- Don't know

I think the GetWitIt notifications are:

- Strongly Frustrating
- Moderatly Frustrating
- Slightly Frustrating
- Neutral
- Slightly Fun
- Moderatly Fun
- Strongly Fun
- Don't know

I think the GetWitIt notifications are:

- Strongly Unpleasant
- Moderatly Unpleasant
- Slightly Unpleasant
- Neutral
- Slightly Pleasant
- Moderatly Pleasant

- Strongly Pleasant
- Don't know

I think the GetWitIt notifications are:

- Strongly Boring
- Moderatly Boring
- Slightly Boring
- Neutral
- Slightly Exciting
- Moderatly Exciting
- Strongly Exciting
- Don't know

How disruptive did you find the application’s notifications?

- Incredibly Disruptive
- Very Disruptive
- Quite Disruptive
- A little Disruptive
- Not that Disruptive
- Not Disruptive at all
- Don't know

How easy was it to reply to events you were invited to?

- Worst Imaginable
- Very Difficult
- Difficult
- Ok
- Easy
- Very Easy
- Best Imaginable
- Don't know

How easy was it to create new events?

- Worst Imaginable
- Very Difficult
- Difficult
- Ok
- Easy
- Very Easy
- Best Imaginable
- Don't know

How easy was it to rate events?

- Worst Imaginable
- Very Difficult
- Difficult
- Ok
- Easy
- Very Easy
- Best Imaginable
- Don't know

How easy was it to know how many events you have gone to in the past?

- Worst Imaginable
- Very Difficult
- Difficult
- Ok
- Easy
- Very Easy
- Best Imaginable
- Don't know

How much information did you have about the invitees and attendance for the events you created?

- Very little Information
- Little Information
- Enough Information
- Don't know

How much information did you have about the invitees and attendance for the events you were invited to?

- Very little Information
- Little Information
- Enough Information
- Don't know

Appendix IV - Zones of Motion joints

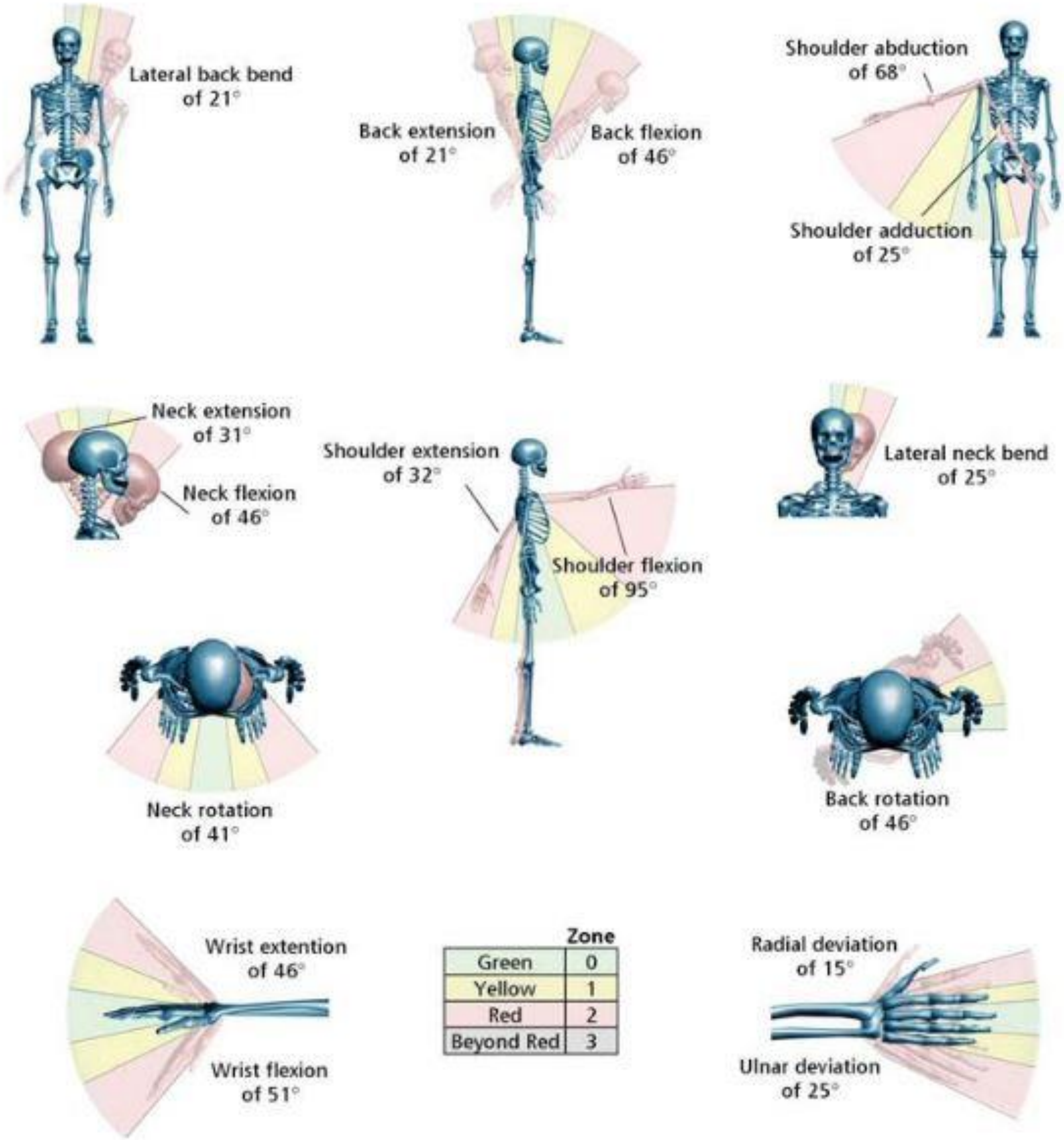

Figure 8.13 Joints motion for the wrist, shoulder, back and neck, adapted from [22]

		Range of Motion Zones			
	Movement	0	1	2	3
Wrist	Flexion	0 – 10	11 – 25	26 – 50	51+
	Extension	0 – 9	10 – 23	24 – 45	46+
	Radial Deviation	0 – 3	4 – 7	8 – 14	15+
	Ulnar Deviation	0 – 5	6 – 12	13 – 24	25+
Shoulder	Flexion	0 – 19	20 – 47	48 – 94	95+
	Extension	0 – 6	7 – 15	16 – 31	32+
	Adduction	0 – 5	6 – 12	13 – 24	25+
	Abduction	0 – 13	14 – 34	35 – 67	68+
Back	Flexion	0 – 10	11 – 25	26 – 45	46+
	Extension	0 – 5	6 – 10	11 – 20	21+
	Rotational	0 – 10	11 – 25	26 – 45	46+
	Lateral Bend	0 – 5	6 – 10	11 – 20	21+
Neck	Flexion	0 – 9	10 – 22	23 – 45	46+
	Extension	0 – 6	7 – 15	16 – 30	31+
	Rotational	0 – 8	9 – 20	21 – 40	41+
	Lateral Bend	0 – 5	6 – 12	13 – 24	25+

Figure 8.14 Measurements in degrees for the movement joints present in Figure 8.13, adapted from [22]

Appendix V - Installation steps for: OpenCV, JavaCV, Java Media Framework and other programs

The instructions to follow to install OpenCV, JavaCV and Java Media Framework are:

1. First make download of the following programs:
   - 1.1. Java™ Platform, Standard Edition Development Kit (JDK™), from http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html (optional, only if you don't have it previously install)
   - 1.2. Java Media Framework, from http://www.oracle.com/technetwork/java/javase/download-142937.html
     - 1.2.1. Install it on the default folder and uncheck the option "Move DLLs to Windows/Systems directory (recommended)", see Figure 8.15

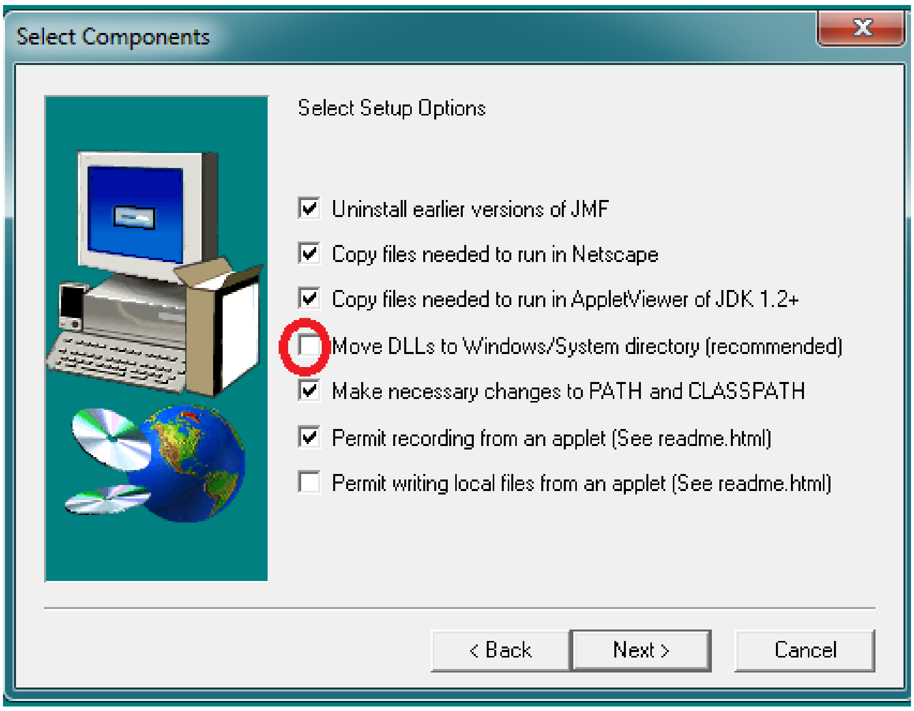

Figure 8.15 Java Media Framework - Select Components window

   - 1.3. OpenCV-2.4.5, from http://opencv.org/downloads.html
     - 1.3.1. Install it in "C:\"
   - 1.4. JavaCV: javacv-0.5-src.zip, javacv-0.5-bin.zip and javacv-0.5-cppjars.zip from, http://code.google.com/p/javacv/
     - 1.4.1. Unzip all the folders using the option "Unzip here" and you will get the following folders: javacv, javacv-bin and javacv-cppjars.

1.4.2. Put the folders javacv and javacv-cppjars into the javacv-bin

1.4.3. Put the javacv-bin into the following path: “c:\”

1.5. RXTX 2.1: rxtx-2.1-7-bins-r2.zip, from http://rxtx.qbang.org/wiki/index.php/Download

1.5.1. Put the file rxtxSerial.dll in the path “C:\Program Files\Java\jdk1.7.0_25\bin”

1.5.2. Put the file RXTXcomm.jar in the path “C:\Program Files\Java\jdk1.7.0_25\lib\ext”

1.6. XAMPP, from http://www.apachefriends.org/pt_br/xampp.html

Appendix VI - Process to detect Motion

In this sub-section it's going to be explained how it was calculated the movement as well the distance of that movement used in the motion detector. The original process developed by Andrew Davison [24] was kept inalterated and follow the next procedure: every time a picture was taken was store in a BufferedImage in order to apply some OpenCV functions (information about this functions was taken from [38]) it was needed first to transform them into IplImage using the convertFrame(BufferedImage buffIm) function. The second phase was apply the function cvAbsDiff(const CvArr* *src1*, const CvArr* *src2*, CvArr* *dst*) where *src1* –array of current image, *src2* –array of the previous image and *dst* –array for destination image. This function calculates absolute difference between two arrays using the Equation 8.1:

$$dst(i)_c = | scr1(I)_c - src2(I)_c | \quad \text{Equation 8.1}$$

The third step its apply the function cvThreshold(const CvArr* *src*, CvArr* *dst*, double *threshold*, double *max_value*, int *threshold_type*) where *src* – input array (single-channel, 8-bit or 32-bit floating point), *dst* – output array of the same size and type as *src*, thresh – threshold value, *maxval* – maximum value to use with the THRESH_BINARY and THRESH_BINARY_INV, thresholding types and *type* – one of the thresholding type and it can be: CV_THRESH_BINARY, CV_THRESH_BINARY_INV, CV_THRESH_TRUNC, CV_THRESH_TOZERO and CV_THRESH_TOZERO_INV. This function is typically used to get a bi-level (binary) image out of a grayscale image or for removing a noise. The fourth step its apply the countNonZero(const MatND& *mtx*) where *mtx* – Single-channel array. This function returns the number of non-zero elements in *mtx*, this means for example movement. The fifth step it's apply the function cvGetSpatialMoment(CvMoments* *moments*, int *x_order*, int *y_order*) where *moments* – The moment state, calculated by Moments, *x_order* – x order of the retrieved moment $-x\_order \geq 0$, *y_order* – y order of the retrieved moment – $y\_order \geq 0$, and $x\_order + y\_order \leq 3$. The function retrieves the spatial moment, which in the case of image *moments* is defined as Equation 8.2:

$$M_{x\_order, y\_order} = \sum_{x,y} \left(I(x,y) * x^{x\_order} * y^{y\_order} \right) \quad \text{Equation 8.2}$$

Where $I(x,y)$ it's the intensity of the pixel (x,y).The sixth step it's to create the Point where the movement was found base on Equation 8.3, Equation 8.4 and Equation 8.5:

$$xPoint = \frac{\text{cvGetSpatialMoment(moments, 1, 0)}}{\text{cvGetSpatialMoment(moments, 0, 0)}} \quad \text{Equation 8.3}$$

$$yPoint = \frac{cvGetSpatialMoment(moments, 0, 1)}{cvGetSpatialMoment(moments, 0, 0)}$$ Equation 8.4

$$Point = (xPoint; yPoint)$$ Equation 8.5

The last step it's to find the distance and the angle formed between the movement detected by the previous picture and the actual one. To collect the distance it was used the Equation 8.6

$$distance = \sqrt{x^2 + y^2}$$ Equation 8.6

Where x and y are given by Equation 8.7 and Equation 8.8:

$$x = x\_actualPoint - x\_previousPoint$$ Equation 8.7

$$y = -1 * (y\_actualPoint - y\_previousPoint)$$ Equation 8.8

To obtain the angle it was made a conversion from the Cartesian coordinates $(x; y)$ to polar coordinates $(r; \theta)$, where r it's the hypotenuse formed by $(x; y)$ and θ it's the angle formed between x and r (see Figure 8.16).

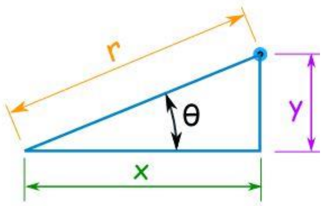

Figure 8.16 Diagram of how Cartesian coordinates relate with Polar coordinates, adapted from [39]

In order to obtain the angle θ it was used the Equation 8.9:

$$\theta = \tan^{-1}\left(\frac{y}{x}\right)$$ Equation 8.9

Appendix VII - Entity–Relationship Model for Pilot Study

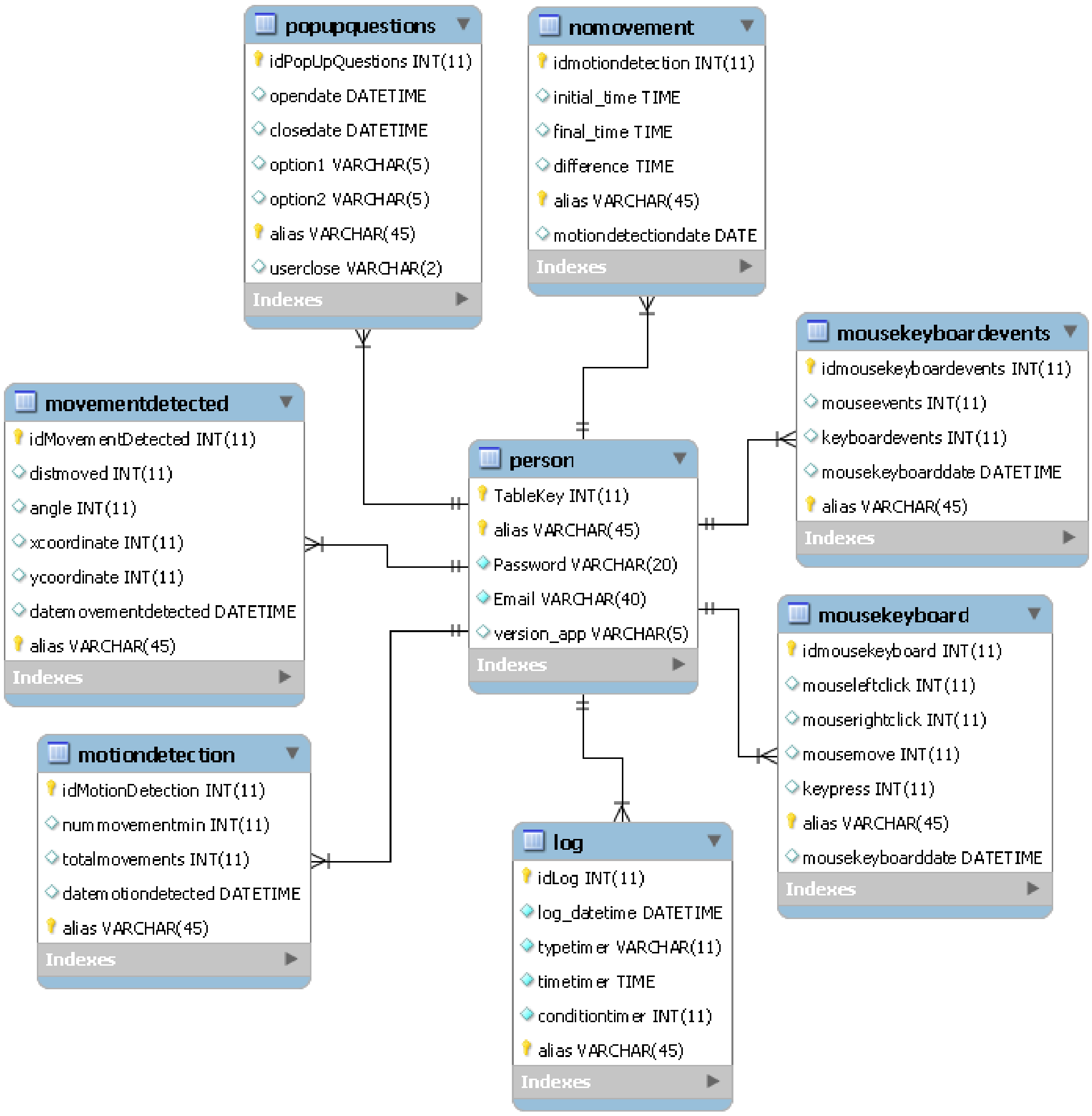

Figure 8.17 Entity–Relationship Model use in Study I

Appendix VIII - Entity–Relationship Model for Study I

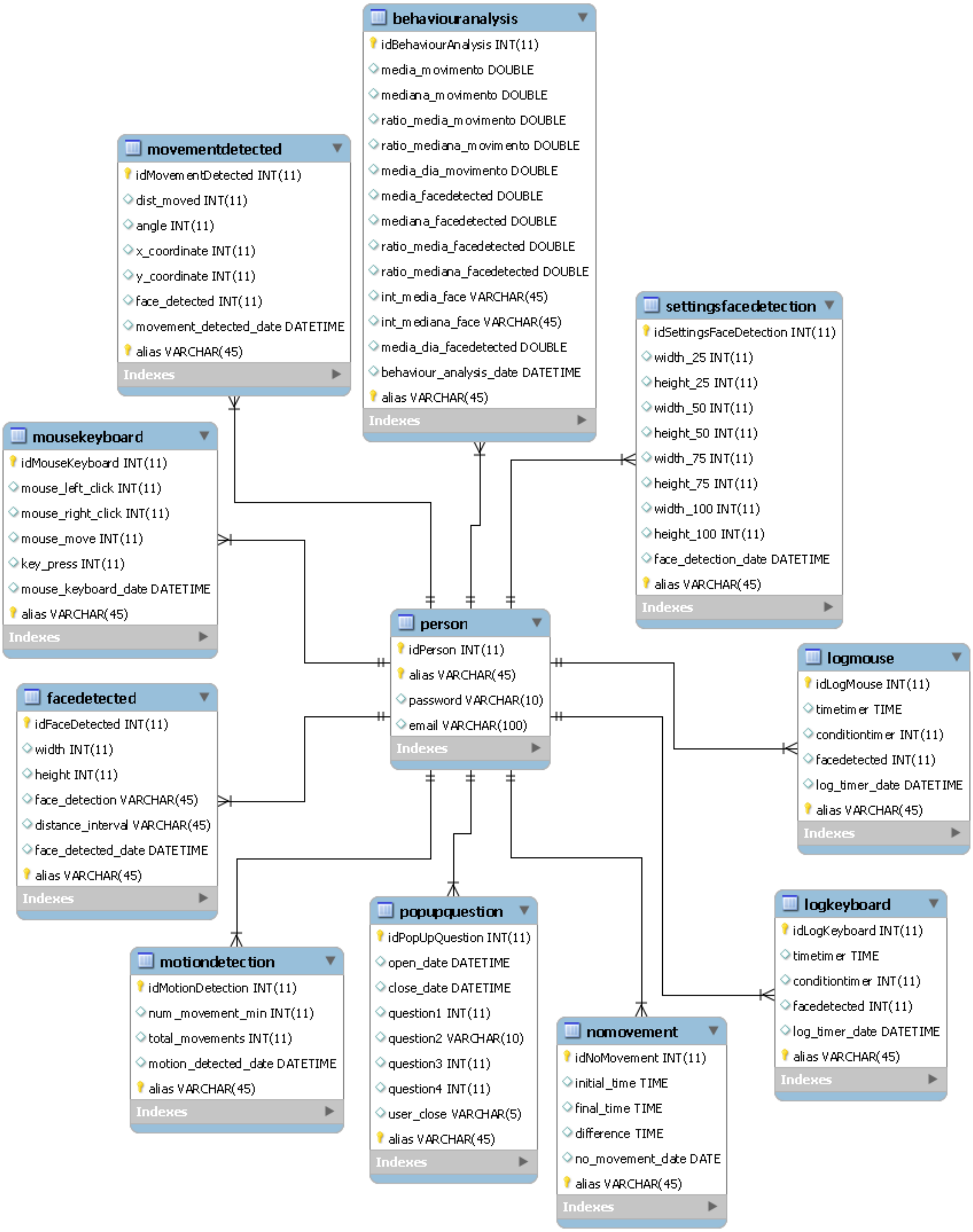

Figure 8.18 Entity–Relationship Model use in Study II

Appendix IX - How to build the origami cubes

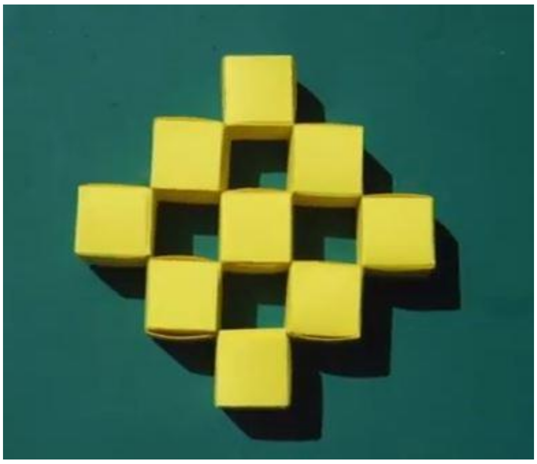

Figure 8.19 Origami of nine cubes, final result

Figure 8.20 Cardboards stripes

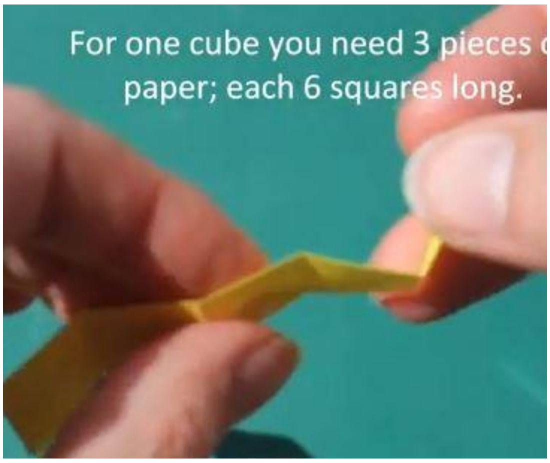

Figure 8.21 What you need to build one cube

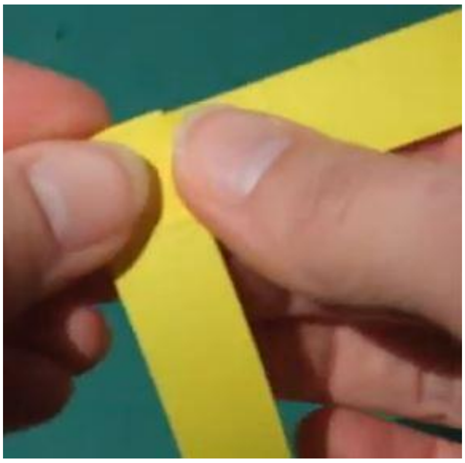

Figure 8.22 Rolling two strips with the same size

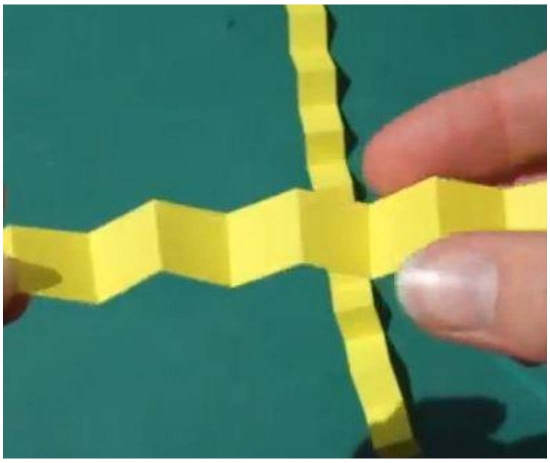

Figure 8.23 Result of rolling two strips

Figure 8.24 Three stripes each with six squares long

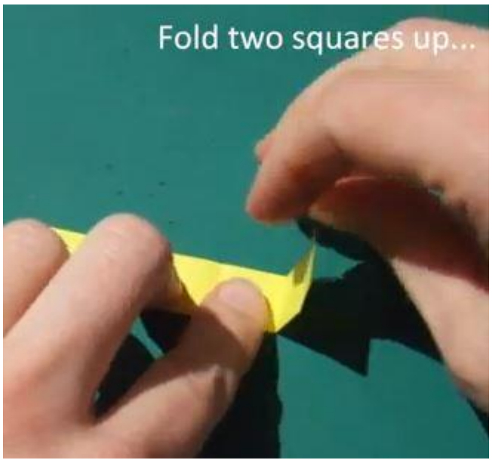

Figure 8.25 With strip one, fold two squares up

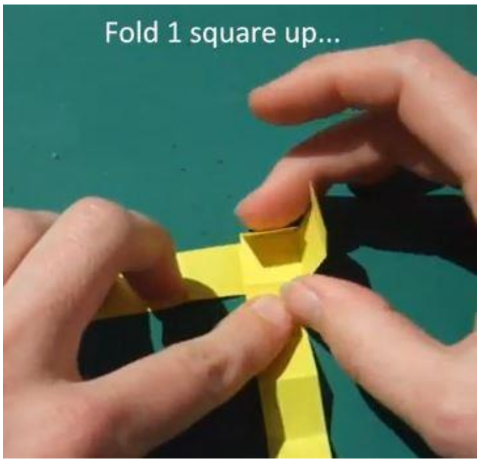

Figure 8.26 With strip two, fold one square up

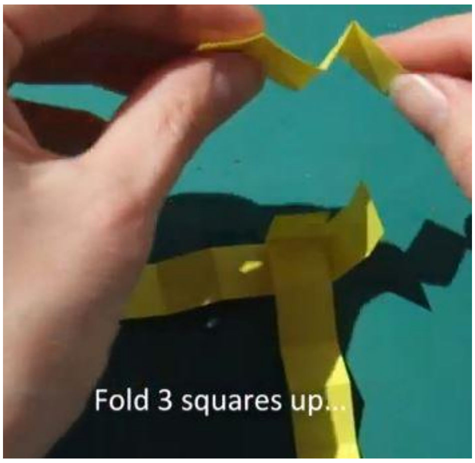

Figure 8.27 With strip three, fold three squares up

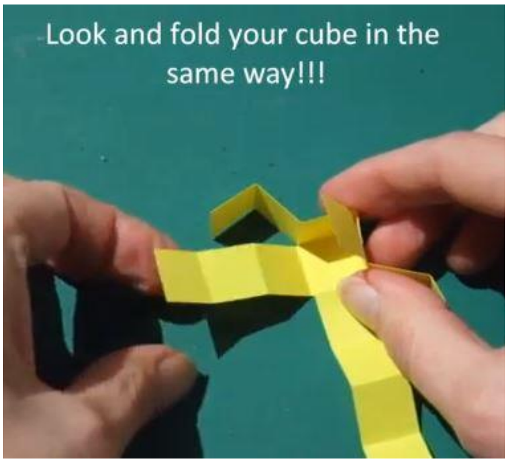

Figure 8.28 Start folding the cubes one the same way with strip three

Figure 8.29 Fold cubes in the same way with one of the remaining strips

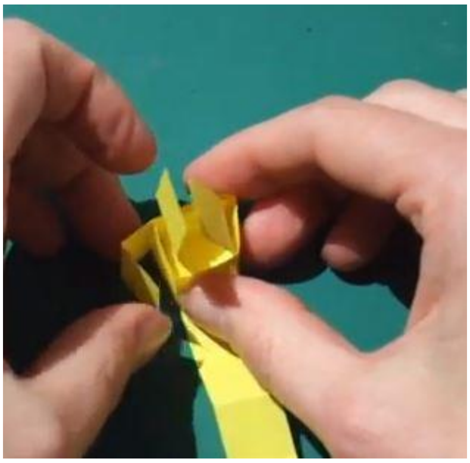

Figure 8.30 Fold cubes in the same way with last strip

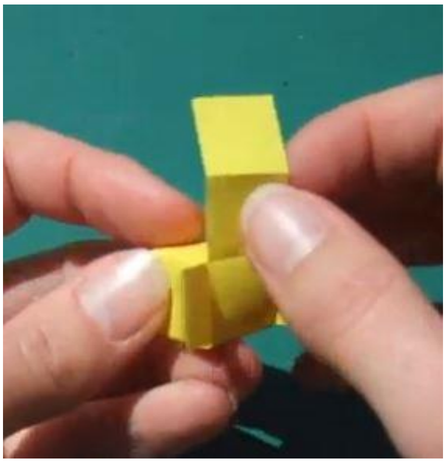

Figure 8.31 Folding cubes where one strip has three last cubes

Figure 8.32 Folding last cube of a strip

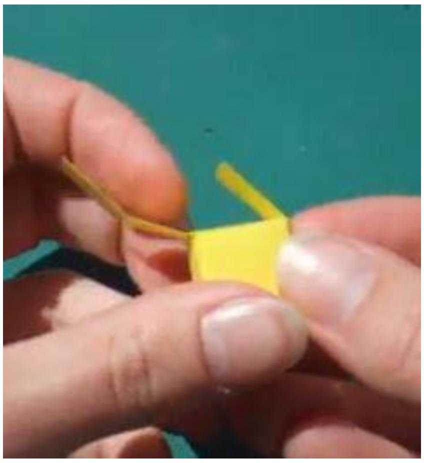

Figure 8.33 Folding cubes of another strip

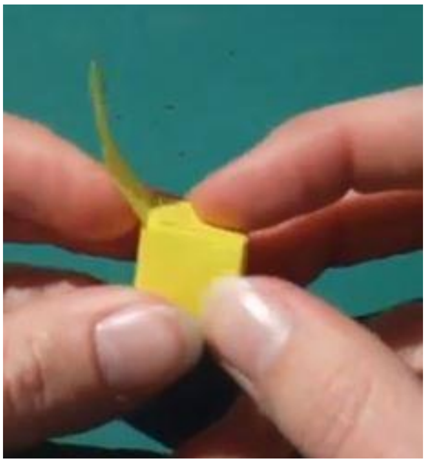

Figure 8.34 Folding cubes were the strip has only two cubes

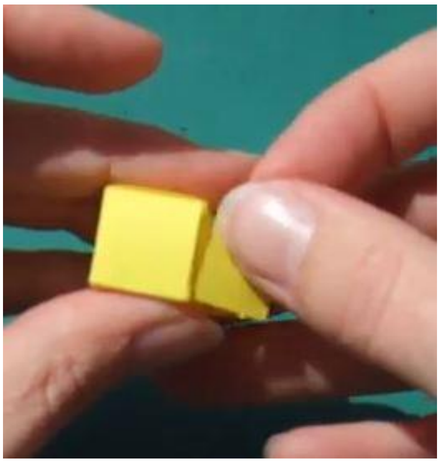

Figure 8.35 Folding last cube

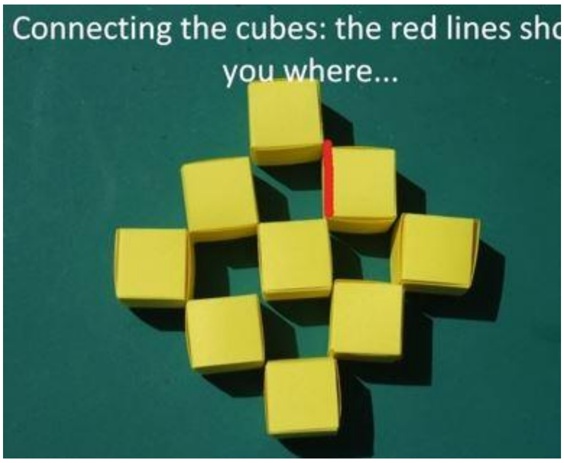

Figure 8.36 Diagram of who to connect the cubes

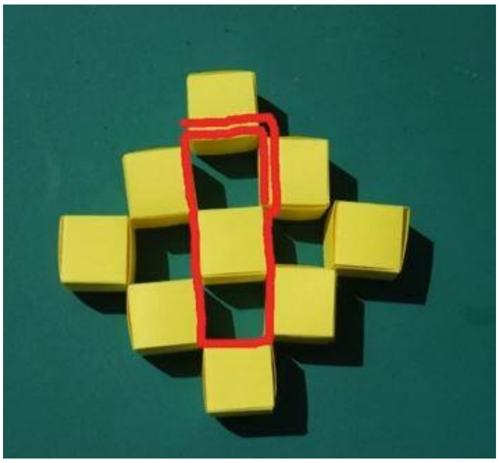

Figure 8.37 How connect the middle cubes

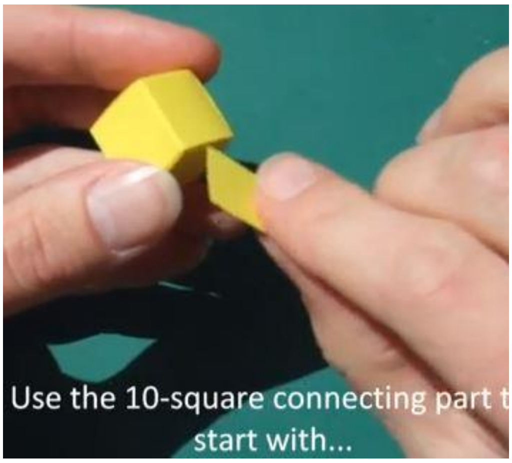

Figure 8.38 Start connecting the cubes with a ten square strip

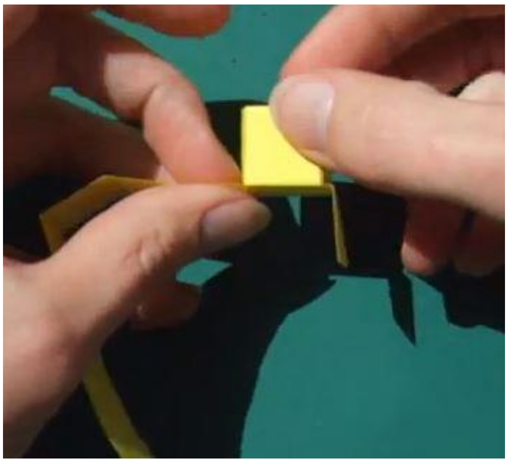

Figure 8.39 Result of the connection with one cube

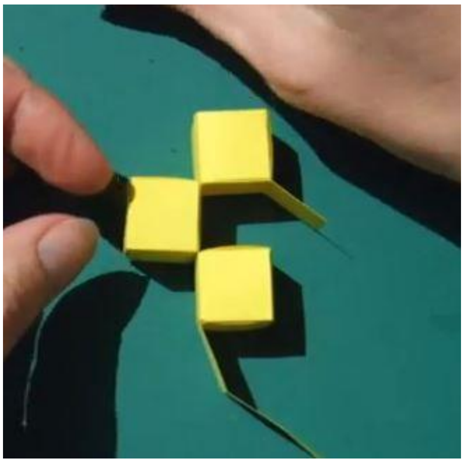

Figure 8.40 Result of the connection with three cubes

Figure 8.41 Result of the connection with six cubes

Figure 8.42 Result of the middle connection

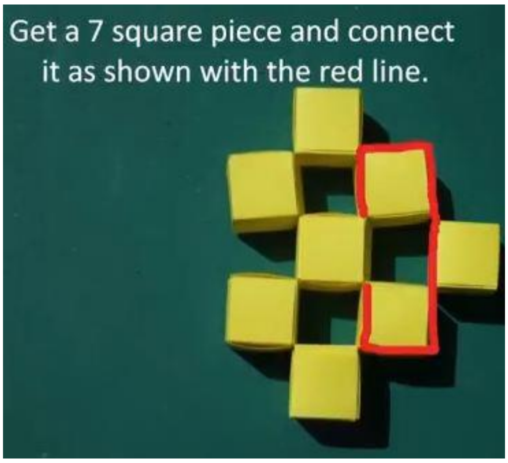

Figure 8.43 How to connect the left side of the cubes to the middle group

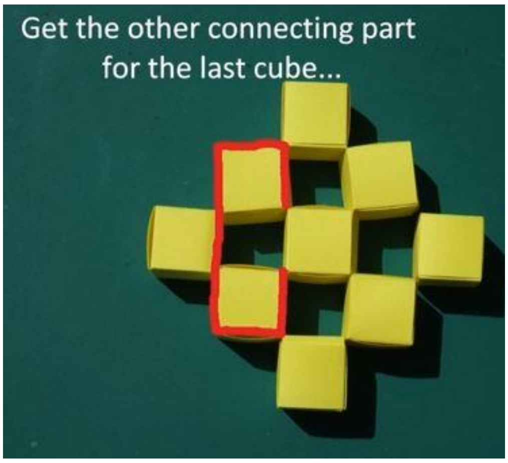

Figure 8.44 How to connect the right side of the cubes to the middle group

All the pictures present in this appendix were adapted from the YouTube video [40]

Appendix X - Schematic from Screen prototype

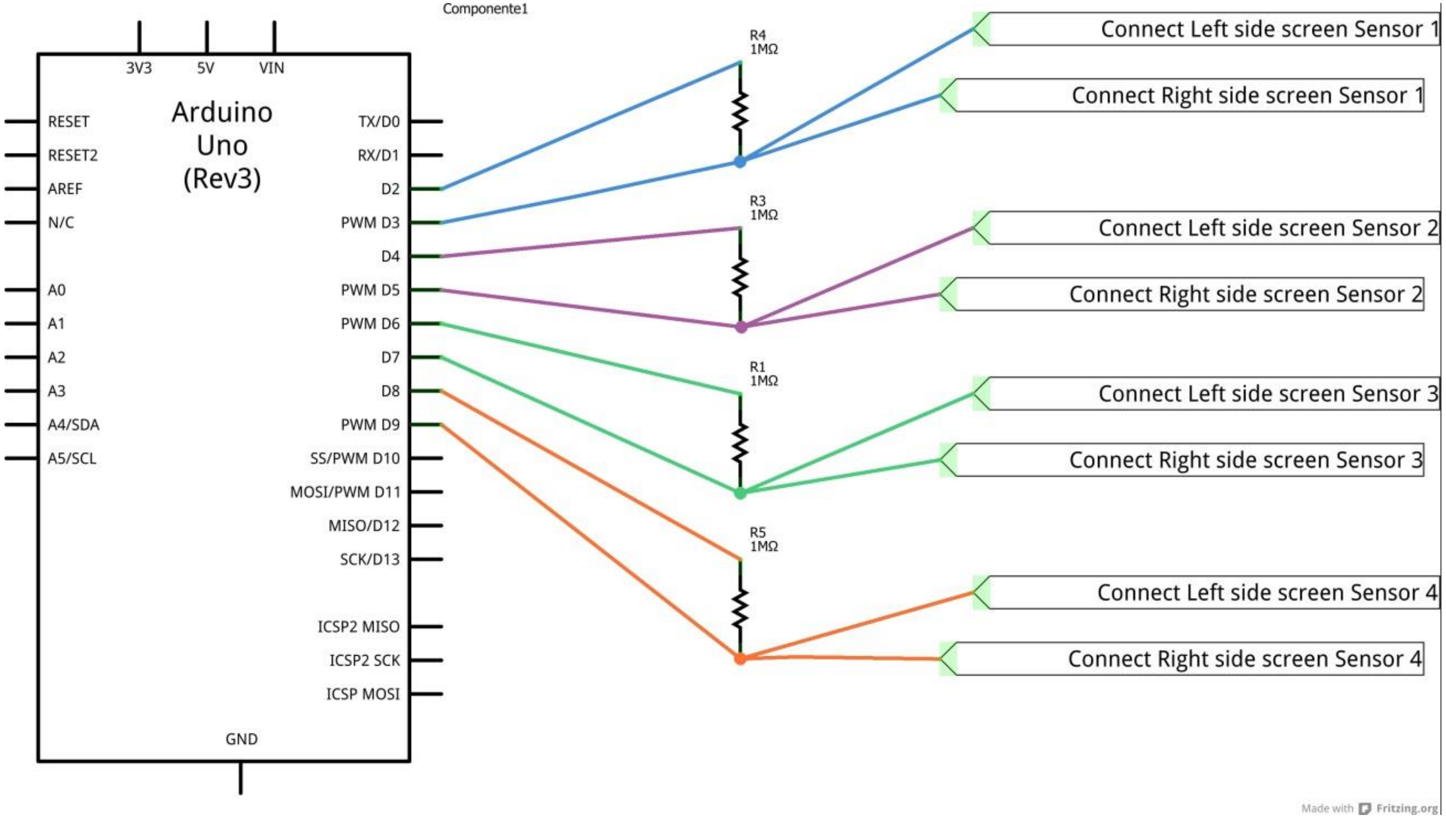

Figure 8.45 Schematic used for screen prototype

Appendix XI - Schematic from Sculpture prototype

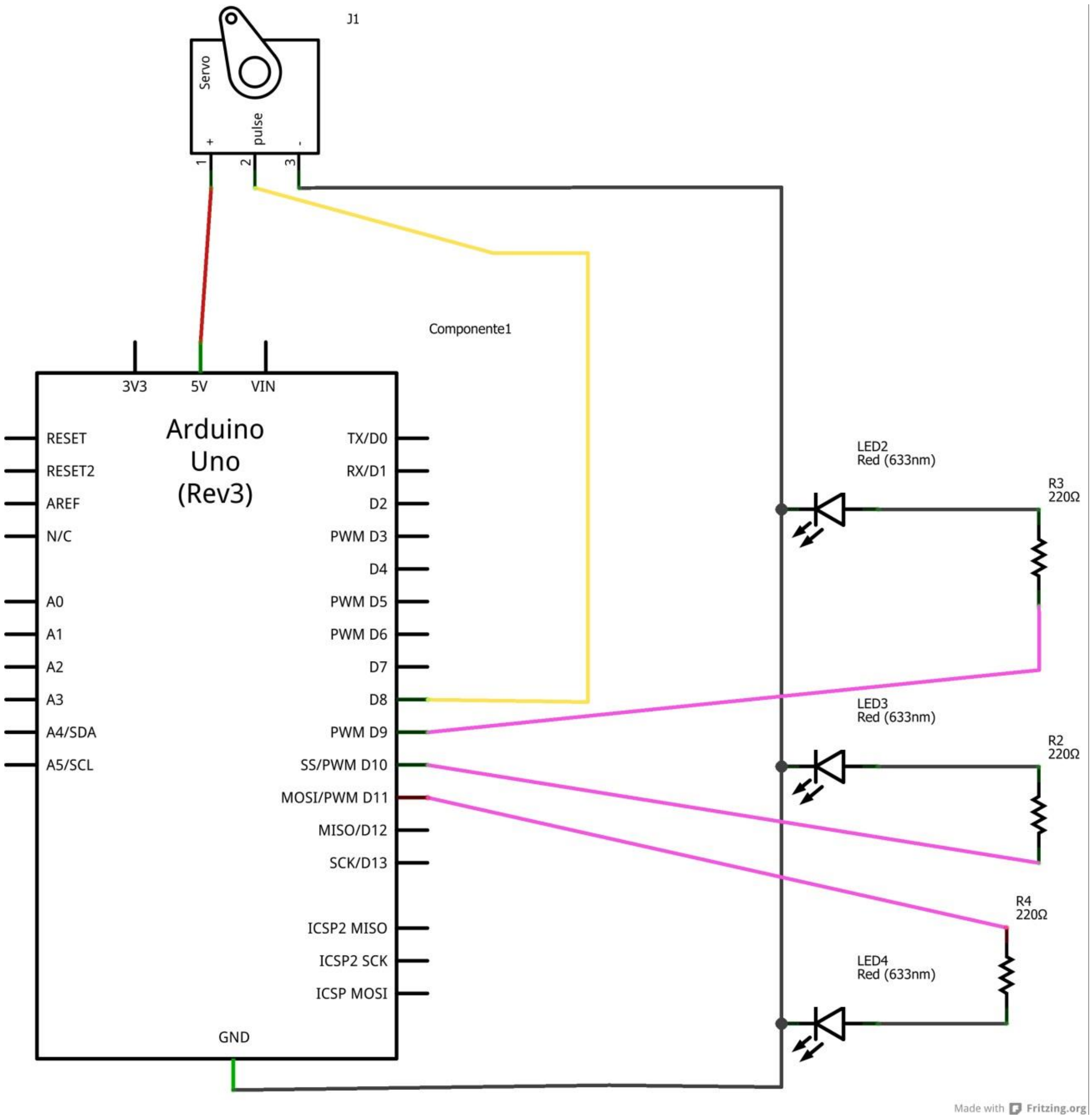

Figure 8.46 Schematic used in sculpture prototype

Appendix XII - Entity–Relationship Model for Final application

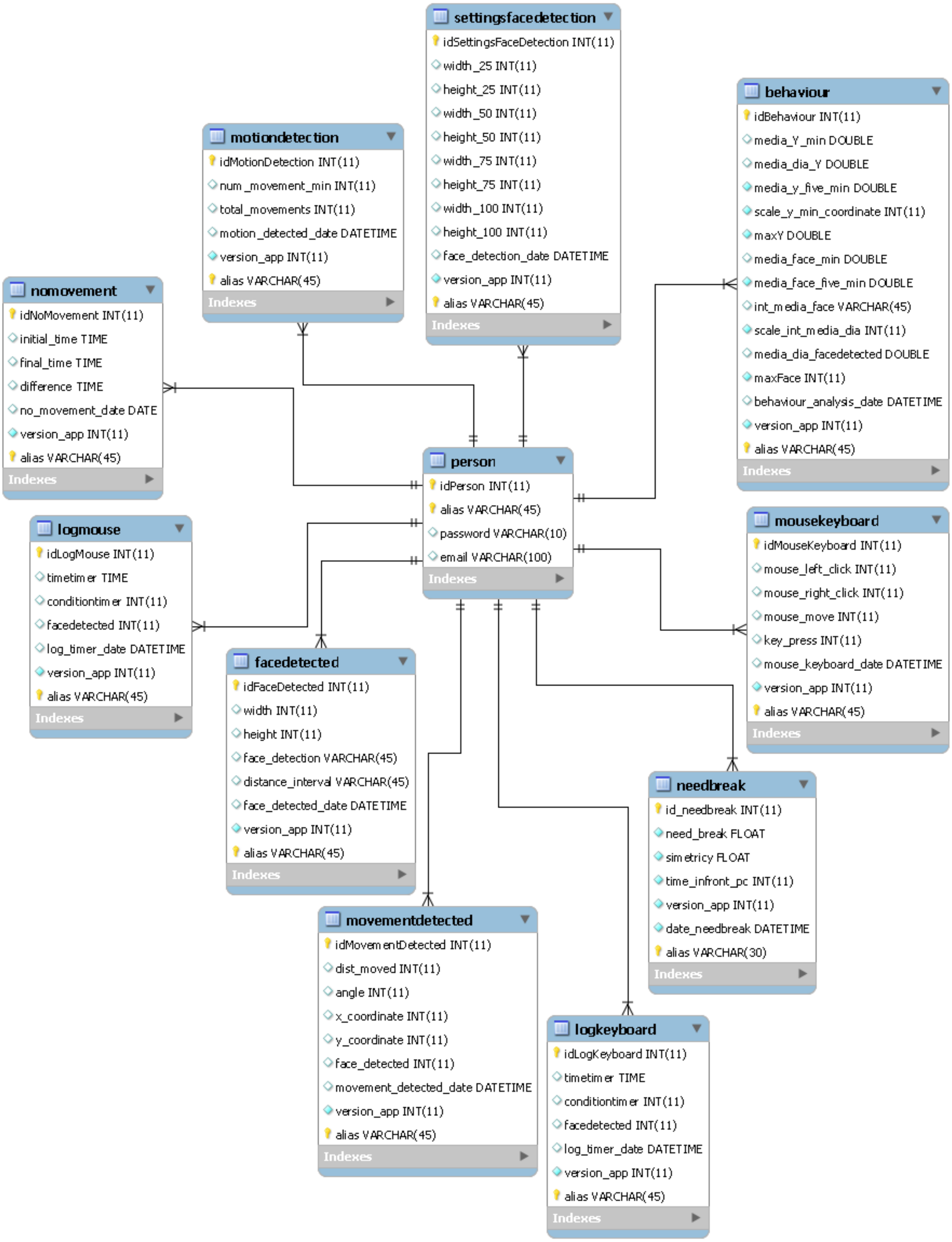

Figure 8.47 Entity–Relationship Model use in Final application

Appendix XIII - Interview protocol for the Final application

Introduction

The participation in the set of interviews that will be made its volunteer, all the information collected will be confidential.

The set will be composed by seven interviews and each will have a maximum duration of twenty minutes. In each interview it will be made specific questions related with the application used in the last two/one day.

Pre-use

In this group of questions we want to understand the number of breaks the use does during the day. To verify this we will observe if the user is using the computer or not. Basically we want to understand his current level of sedentary behaviour.

The information collected it will be compered posteriorly with the two weeks of use with the application, to see id was any influence and impact of each version.

1. In a scale of 1 to 5 how long you spend your day sit? (1 less frequent and 5 much frequent)
   1.1. In a scale of 1 to 5, how much your daily work depends of the computer? (1 less frequent and 5 much frequent)
2. When you use to do breaks?
3. In a scale of 1 to 5 how many breaks you do daily? (1 less frequent and 5 much frequent)
4. With what duration?
5. For you what is a break?
6. You use to do breaks at the computer?

Between different versions

We want to understand if the system had any influence in the users – if they did more breaks or not and we want to understand the impact of each version.

Wallpaper

1. Did you notice any change in your breaks compared with your regular ones? If yes, what change?

Do this question only if the user says yes to the previous one.

   1.1. In a scale of 1 to 5, how far the system was responsible for this change? (1 less responsible and 5 much responsible)

1.1.1. If the previous answer was between 3 and 5, in what way influenced?

Do a recreation of the wallpapers changing and ask what they felt about each change. (See if the user had the notions of posture and the need of doing a break.)

Sculpture

1. Did you notice any change in your breaks compared with your regular ones? If yes, what change?

Do this question only if the user says yes to the previous one.

1.2. In a scale of 1 to 5, how far the system was responsible for this change? (1 less responsible and 5 much responsible)
   1.2.1. If the previous answer was between 3 and 5, in what way influenced?

2. Do you felt that the system had influence in the surrounding persons? If yes, what impact had?

Do a recreation of the sculpture changing when recommending a break and when it's not. Ask what they felt about each change. (See if the user had the notions of posture and the need of doing a break.)

Screen

1. Did you notice any change in your breaks compared with your regular ones? If yes, what change?

Do this question only if the user says yes to the previous one.

1.1. In a scale of 1 to 5, how far the system was responsible for this change? (1 less responsible and 5 much responsible)
   1.1.1. If the previous answer was between 3 and 5, in what way influenced?

2. Do you felt that the system had influence in the surrounding persons? If yes, what impact had?

Do a recreation of the popup recommending a break when show to the user for the first time and after the user asks for delay. Ask what they felt about each change. (See if the user had the notions of posture and the need of doing a break.)

Sculpture and Screen

1. Did you notice any change in your breaks compared with your regular ones? If yes, what change?

Do this question only if the user says yes to the previous one.

2.1. In a scale of 1 to 5, how far the system was responsible for this change? (1 less responsible and 5 much responsible)

2.1.1. If the previous answer was between 3 and 5, in what way influenced?

3. Do you felt that the system had influence in the surrounding persons? If yes, what impact had?

Do a recreation of the popup + movement of the sculpture recommending a break when show to the user for the first time and after the user asks for delay and the sculpture movement when not recommending a break. Ask what they felt about each change. (See if the user had the notions of posture and the need of doing a break.)

Post-use

We want to understand if after given several forms of feedback to the user if we take them all they will fell any difference.

Without feedback

1. Did you notice any change compared with the last two weeks? If yes, what changed? (We want to understand if taking the feedback of the application there was any change in the user's behaviour – negative and/or positive).

If the user answers yes to the previous question:

2. In what way the system made you miss? (For example: try to see if the feedback present in the app was important or not).

Comparison questions

We want to understand which version of feedback the users prefer the most and the most easily to understand the feedback.

1. In the past two weeks, you used four different ways of receiving feedback – wallpaper, sculpture, screen and sculpture + screen – in your opinion which of this conditions provided a more intuitive/easy way to understand the feedback?
   1.1. Why this condition was more significant to you?