

Master Thesis

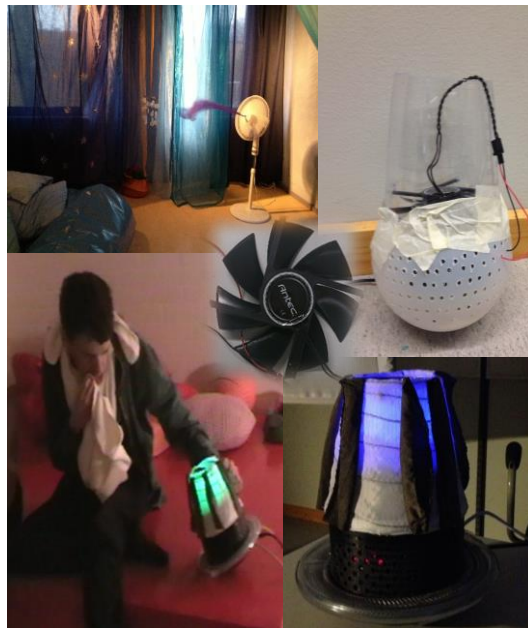
Developing interactive interfaces for people with developmental disabilities to be used in Snoezelen environments

Charlotta Viklund & Pamela Svensson

Certec, Division of Rehabilitation Engineering Research

Department of Design Sciences

Faculty of Engineering LTH • Lund University • 2014



LUND UNIVERSITY

Developing interactive interfaces for people with developmental disabilities to be used in Snoezelen environments

Charlotta Viklund & Pamela Svensson

Certec, Division of Rehabilitation Engineering Research • Department of Design Sciences

Faculty of Engineering LTH • Lund University • 2014

Certec, Division of Rehabilitation Engineering research
Department of Design Sciences
Faculty of Engineering LTH, Lund University
P.O. Box 118
SE-221 00 Lund
Sweden

Certec report 2:2014

Preface

This master thesis would not have been possible without the support and feedback from many people.

We would like to express our gratitude to our supervisor, Héctor Caltenco. Thanks for all the support and useful advices about both technical issues as well as design issues. Your engagement and encouragement has inspired us to develop a great design. We will never forget, “keep it simple”. Thanks for your patience and your help to structure our, sometimes, disorganized thoughts and for the useful comments and feedback on the report.

Thanks to Henrik Svarrer-Larsen for the inputs on how to perform and think in a design process, and for the feedback on the report. Thanks for opening up our eyes to new things and making us question our work.

We want to thank all the amazing participants at the Snoezelen institutions at *Safiren*, Malmö, and *Snoezelhuset*, Gentofte and the staff Lone Johansson, Lone Boel and Marian Fyrkilde. Your participation made this project possible. We also want to thank the staff, Ellen Westberg and Eva Widerdal at DUMLE, Lund, for all the feedback and the positive energy you have given us.

Lund, March 2014

Charlotta Viklund and Pamela Svensson

Abstract

Much of the equipment used in Snoezelen environments today is not interactive, and the equipment that is interactive can be hard for the user to access. By having interactive equipment in Snoezelen environments the user gets to experience and hopefully better understand the concept of cause and effect.

The purpose of this master thesis has been to develop an interactive interface that can appeal to a wide range of users. The target group has been adults with developmental disabilities and sensory deprivation. By a small effort they should be able to get a response from the interface.

To do this different methods have been used, such as qualitative research methods and bodystorming. The work was developed following a user-centered design method and an iterative design process where Snoezelen users from two different Snoezelen institutions in the Öresund Region participated. There were four visits to each place to get to know the environment, the staff and the users, and to perform tests with the users and the interactive interface.

The interactive interface is called Glownado. It is quite small and allows the user to interact in different ways, it can be approached, put into motion, and it has different tactile structure that the user can touch. This makes it easy to interact with Glownado, in a way that suits the user's needs and abilities, which makes Glownado fit a wider range of users. The feedback the user gets is in form of wind, sound and light, and creates a multisensory stimulation localized to one area, which reduce confusion for the user. Depending on how the user interacts with it, it has different behaviors to respond to the specific action. It was important that the user got an immediate response to their action, to understand that they were the ones making it happen. Wind has never been implemented in interactive Snoezelen equipment before. It created a surprise moment, which made the users curious to continue explore.

Keywords: Snoezelen, Interactive Design, Arduino, Qualitative Research Methods, Multisensory Stimulation

Sammanfattning

Detta examensarbete har gått ut på att skapa ett interaktivt interface, som kan användas i Snoezelen miljöer.

I dagens Snoezelen-institutioner finns det få interaktiva föremål. De interaktiva föremål som finns, kan ibland vara svåra för användaren att interagera med. Många av föremålen kräver till exempel att användaren har förmågan att både kunna förstå och trycka på en knapp. Genom interaktivitet kan användaren få uppleva att det är de som påverkar det som händer runtomkring. Målet har varit att utveckla ett interaktivt interface som ska vara lättillgängligt för användare. Genom en liten rörelse ska de kunna få en respons från föremålet. Detta gör det möjligt att nå en större grupp som kan dra nytta av interagerandet. Det interaktiva interface som utvecklades i projektet fick namnet Glownado.

Målgruppen har varit vuxna som har olika typer av utvecklingsstörning. Projektet gjordes i samarbete med tre olika Snoezelen-institutioner, där fem olika personer som besöker institutionerna har medverkat.

Projektet utgjordes av en iterativ designprocess. Där Glownado utvecklats genom bodystorming, feedback, från handledare och personal på Snoezelen-institutioner samt, från observationer där användarna interagerande med Glownado. En kvalitativ metod har använts för att dokumentera och analysera observationerna.

Designens utformning och de sensorer som användes, gjorde det möjligt att skapa ett interaktivt interface som var tillräckligt känsligt för att reagera på nätta rörelser men samtidigt tålig för att klara av tuffa rörelser. Detta öppnar upp möjligheter för användaren att interagera med så få begränsningar som möjligt. Genom att skapa graduell respons, som följer användarens beteende, och samtidigt ge snabb respons på handlingen får användaren uppleva orsak och verkan.

En fläkt implementerades för att ge feedback på användarens interaktion. Den plötsliga starka vinden bidrar till ett överraskningsmoment genom dess osynliga taktila återkoppling. Detta skapar spänning och nyfikenhet hos användaren att fortsätta utforska och interagera. Ljus och ljud implementerades också för att ge återkoppling till användaren, men även för att skapa ett intresse hos användaren att börja interagera. Dessa gemensamma intryck kan tillsammans stimulera flera sinnen och bidra till ett välbefinnande.

Under projektets gång dök det upp faktorer som kom att påverka interaktionen och beteendet. Idéer på hur Glownado kan vidareutvecklats har därför föreslagits.

Table of Contents

1 Introduction	1
1.1 Senses	1
1.2 Multisensory Stimulation	1
1.3 Developmental Disabilities and Sensory Deprivation	2
1.4 Snoezelen	3
1.5 The SID project	5
1.6 Aim	5
2 Method	7
2.1 User-centered Design	7
2.2 Research Methods	7
2.3 Bodystorming	8
2.4 Risk Analysis	8
2.5 Project Structure	9
2.5.1 Vertical and Horizontal Prototyping	9
2.5.2 Structured Design	9
2.5.3 State Machines	10
3 Pre-study	13
3.1 Snoezelen Institutions	13
3.1.1 Snoezelhuset, Gentofte	14
3.1.2 Safirens Upplevelsehus, Malmö	15
3.1.3 Hjälpmedelsverksamheten, Lund	16
3.2 Participants	17
3.3 Interactive Snoezelen Equipment	18
4 Proposed Solutions	21
4.1 Brainstorming	21
4.2 Interactive Bubble Tube	22

4.3 Glownado	24
4.4 Concept selection.....	24
5 Design Implementation	27
5.1 Overview, Criteria and Structured Design.....	27
5.2 Physical Material Development.....	30
5.2.1 Implementing the Design	30
5.2.2 Implementing Wind	34
5.2.3 Implementing Light.....	37
5.2.4 Implementing Sound	37
5.2.5 Implementing Control Box.....	39
5.3 Interface Behavior, part 1	40
5.3.1 Sleep mode.....	41
5.3.2 Active mode	42
5.3.3 Observations in Snoezelen environment, part 1	42
5.4 Interface Behavior, part 2.....	44
5.4.1 Sleep mode.....	45
5.4.2 Active mode	45
5.4.3 Wind mode.....	46
5.4.4 Sound mode.....	47
5.4.5 Observations in Snoezelen environment, part 2	47
5.5 Interface Behavior, part 3.....	48
5.5.1 Sleep mode.....	48
5.5.2 Active Mode	48
5.5.3 Wind mode.....	49
5.5.4 Sound mode.....	49
5.5.5 Other Features.....	49
5.5.6 Observations in Snoezelen environment, part 3	49
5.6 Interface Behavior, part 4.....	51
5.6.1 Sleep mode.....	53
5.6.2 Active mode	53
5.6.3 Sound mode.....	53
5.6.4 Other Features.....	54
5.6.5 Observations in Snoezelen environment, part 4	54
6 Conclusions and Future Work.....	57
6.1 Conclusions.....	57

6.2 Future Work.....	60
6.2.1 Improvement of the Capacitive Sensor	60
6.2.2 Reducing Vibrations.....	60
6.2.3 Controlling the Wind.....	60
6.2.4 Implementing a Scent	60
References	63
Appendix A : Materials	67
A.1 Controller	67
A.2 Capacitive sensor	67
A.3 Accelerometer.....	68
A.4 Other Sensors.....	69
A.5 Light	70
A.6 Sound.....	70
A.7 CAD/3D Printing	70
A.8 Conductive Fabric.....	70
A.9 Other Materials	70
Appendix B : Tables	73
B.1 Fans.....	73
B.2 Risk Analysis	73
Appendix C : Hardware and Software	75
C.1 Schematic.....	75
C.2 Code	76
C.2.1 Master	76
C.2.2 CapSenseValue	80
C.2.3 Wind	81
C.2.4 Light	82
C.2.5 Modes.....	86
C.2.6 SenseMode	86
C.2.7 Sound.....	90

1 Introduction

This chapter includes the concept of senses and multisensory stimulation. It also includes an introduction to developmental disabilities. It provides background information about Snoezelen and SID and why it is beneficial to implement interactive design at Snoezelen institutions. The chapter also presents the aim of the master thesis.

1.1 Senses

Aristotle identified five senses in human beings, sight, hearing, smell, taste and touch, but there is no exact number of how many senses the human being has. The sense of balance, for example, is another type of sense. Nowadays the senses are put in order according to the individual, depending on how the world is being experienced by the individual through the senses (Pagliano, 2012, p.3, 6-7).

Through several sense receptors, the information about what happens inside and outside the body reaches CNS (Central Nervous System). There are several receptor types that respond to different kinds of stimulations, such as smell, touch and hearing. The receptors are divided into two classes, interoception and exteroception. Interoception is when the senses stimulate the receptors inside the body, for example, receptors that exist in the lungs for breathing. These are senses that work on an unconscious level. Exteroception is when stimulation comes from outside the body. This kind of senses are more important for interaction with the environment, since they provide information on a conscious level, as opposed to interoception (Pagliano, 2012, p.12, 16). Senses of exteroception are further divided into two groups, remote and contact senses. The remote senses are senses that can perceive information at large distances, such as sight and hearing. Whereas the contact senses are senses that only can perceive information at close distance, such as touch, smell, taste and a sense of balance (Mertens, 2008, p.16). This is just a rough categorization since there are some senses that can be classified into both groups, such as smell where it is possible to notice the smell far away as well as close.

1.2 Multisensory Stimulation

To gain useful information about the world, the human brain needs to obtain many impressions with the help of many different senses. Pagliano (2012) states “We are multisensory beings, and we live in a multisensory world”.

The surrounding environment provides multiple impressions at the same time. To process all the information, many senses are being activated to create a manageable picture of what is happening. This means that different senses, such as sight, hearing,

smell, taste, touch and a sense of balance needs to merge information in order to perceive the world in an accurate way. Multisensory stimulation is the process when multiple senses are stimulated at the same time and is necessary to get a better understanding of what is happening in the world. Although, senses are necessary for the human being to understand the world and the surrounding, it will not always be regarded as positive or as a nice experience for people with sensory deprivation. Instead, the multisensory stimulation can be frightening, have a lack of meaning or not even being noticed. When having a sensory deprivation, some senses will get more activated while others will get suppressed. When experiencing a situation where the impaired sense will activate, it might appear frightening since such sense is not usually used (Pagliano, 2012, p.6-11, 47, 53). For example, when faced to something unfamiliar, one can feel insecure and lack confidence.

All the stimulation that is being processed contributes to the brain's development, which means that the brain is changing itself when processing different impressions. This is called neuroplasticity. Every sense activates different parts of the brain. When a part of the brain is not being stimulated, because of deprivation, the area associated to the sense might shrink or it can even be taken over by another part that is more active. The brain has an amazing ability of self-repair that is why it is important to learn how to use the senses to minimize the outcome of sensory deprivation. This can be done, by being exposed to multisensory stimulation in a controlled environment (Pagliano, 2012, p.5, 9-10).

1.3 Developmental Disabilities and Sensory Deprivation

Developmental disability is a wide term, which describes many different chronic disabilities that can be cognitive, physical or even both (American Association on Intellectual and Developmental Disabilities, 2013). Cognitive disabilities are also a wide term and the definition varies, but it is commonly defined as, having problem with mental tasks and how to process them (National Center on Accessible Instructional Materials, 2013).

Since the early 1900s, there has been a scientific research on how the brain is being negatively affected when there is a lack of sensory stimulation. Sensory deprivation affects the human brain and can occur from abuse or neglect, or by severe sensory, cognitive and learning disabilities. As long as some part of the human brain is not stimulated, the specific part will wither. This can lead to rigorous life conditions in a way that it will affect the physical development, social and emotional functioning, the behavior, as well as how to communicate and how to learn (Pagliano, 2012, p.6-8).

Developmental disabilities, such as Down syndrome, fetal alcohol syndrome, severe autism, cerebral palsy, etc., can affect cognitive ability, physical functioning, and learning (Centers for Disease Control and Prevention, 2013). Children and adults with developmental disabilities will commonly be affected by sensory deprivation (Pagliano, 2012, p.8).

There are a lot of different disabilities that can be developed at any time during the developmental period. It may arise before birth, when there are many factors influencing such as the genetics, or the health or the behavior of the parent. There can be complications during birth and after birth caused by injuries, infections, etc., which can lead to brain damages (Centers for Disease Control and Prevention, 2013).

1.4 Snoezelen

Individuals with developmental disabilities are often limited in understanding and controlling the environment around them, which often leads to them feeling insecure and becoming withdrawn and, thus, more helpless than they usually are with their disabilities. To minimize this, there were two psychologists working in the USA, Cleland and Clark, who developed a “sensory cafeteria” for people with intellectual disabilities (Pagliano, 2012, p.10). This was further developed by Hulsegge and Verhaul, to occupy the participants with intellectual and multiple disabilities. The environment where the participants experienced sensory stimulation was called Snoezelen and became a registered trademark. In the multisensory environment the individuals are supposed to experience having control over their surroundings and also get a feeling of independence (Mertens, 2008, p.11, 31).

The word Snoezelen is composed of the Dutch words *snuffelen*, to sniff or to snuffle, and *doezelen*, to doze or to snooze. In the Snoezelen institutions the rooms are custom designed for children or adults with developmental disabilities, and the environment is stimulating but peaceful. The idea with Snoezelen is that the users are supposed to stimulate their senses in a non-demanding environment (*snuffelen*). However, it is also important that a user that is over-stimulated, at start, can be calmed down by the environment (*doezelen*). Another advantage with this kind of environment is that it creates a relationship between the users, the environment and the carer. Snoezelen is not only used for people with developmental disabilities they are also used by rehabilitation centers, nursing homes, special needs schools etc., its target groupe is both people with disabilities as well as those without disabilities (Mertens, 2008, p.6, 10, 29). In this project the focus has been people with developmental disabilities.

The Snoezelen rooms have different themes and the equipment can be moved in or out of the rooms. This is to be able to control the amount of stimuli to fulfill the needs of the user. According to Lone Johansson, employee at a Snoezelen institution in Malmö, colors are chosen carefully, since different colors enhance different moods, e.g. enhances activity or calmness. The rooms have speakers to play music, light sources for different visual effects, and a variety of fabrics with tactile structure. The combination of these impressions has a positive impact and helps to calm down and can emphasize memories for the user (Mertens, 2008, p.11). Ball pool is a common element in Snoezelen environment, which provides plenty of tactile stimulation. The purpose of the balls is to give a light pressure against the body to give the user a better sense of where the body starts and ends. To stimulate the sense of balance a Snoezelen room contains, for example, waterbeds, hammocks and beanbags (Mertens, 2008, p.19). They also contain mirror balls, fiber optics, liquid projectors, black lights, etc., which are used to create visual effects.

One of the basic rooms is the white room, figure 1.1. The walls and the ceiling are painted white to enhance the visual effects (Mertens, 2008, p.45). In a white room it is easy to add color with different fabrics or light.



Figure 1.1 A white room with a waterbed and an image from a liquid projector on the wall (SID, 2014).

Another basic room is a dark room, figure 1.2. It has black walls, floor and ceiling, and contains colored and black light to stimulate the sense of sight. This room is especially good for users with visual impairment, since it helps the user focus on the objects by having as few distractions as possible (Pagliano, 2012, p.71).



Figure 1.2 A dark room with a ball pool and strands that lights up by black light (SID, 2014).

1.5 The SID project

SID (Sinnlighet, Interaktion och Delaktighet translated Sensuousness, Interaction and Participation) is a project that took place in three Snoezelen institutions, in Lund and Malmö, Sweden, and Gentofte, Denmark. This was a cooperation between Certec (Department of design Science, University of Lund), FUB (Förening för barn, unga och vuxna med utvecklingsstörning, translated Association for children, youth and adults with developmental disabilities) and Furuboda KompetensCenter, and started 1st July 2010 (SID, 2014). SID's main objective was to arouse the curiosity of the user to explore the Snoezelen environment. This was done by modifying the existing objects and developing new interfaces. The new interfaces were designed to contain its own essence i.e. making them have some agency, but still be manageable, meaningful and comprehensible which are the three components of the "sense of coherence" (Buch, 2006). This is a theoretical formula, which explains how the human function in the role of stress, coined by a professor in medical sociology, Aaron Antonovsky (Buch, 2006). To avoid arousing stress there has to be some coherence in the user's actions with the interface and how the interface responds. It is important that the interface will attract the user to interact with it and arouse the curiosity to continue exploring. Making the Snoezelen environment interactive will hopefully engage the users to explore the outside world as well, since they might understand that they are making an impact on the surroundings. With the SID-project the idea is to use the Snoezelen practice, but with interactive techniques where the users can use their bodies and senses to interact with the environment. The interactive interfaces should give the user a direct response, but also a gradual response to the users' actions. It is important that the interface is predictable but also has a small amount of unpredictability. This is to make it understandable, yet give it some agency, and thus arousing the curiosity of the user (SID, 2014).

1.6 Aim

The aim of this project is to design one or more interactive interfaces that can be used in Snoezelen institutions. It should be manageable, meaningful and comprehensible. This makes it less confusing and reduces the insecurity of the user, which leads to exploration of senses. The interactive interface is supposed to have a gradual response and stimulate different senses. It is also important to arouse the curiosity of the user to interact with the interactive interface, and that the users understand that their actions are the cause of the response given by the interactive interface. With the design it is important that the interactive interface helps the user to snuffle or to doze.

The users who have participated in this study have a variety of abilities and disabilities. The goal has therefore been to make an interface that appeals to a wide range of users.

2 Method

This chapter describes the methods that were used in the project. The project follows a user-centered design process. To simulate the users' perception, bodystorming was used as the main method. During the development of the project there are several factors that can emerge, that might affect negatively during the use, or can cause damage to the interface or the user. Using a risk analysis, these factors can be reduced or even avoided. Other methods used are necessary to follow a structure, keep track of the process, and how to investigate the observations made in the research.

2.1 User-centered Design

In the design process it is important to have the users' perspective in mind (Preece, Rogers and Sharp, 2007, p.5), in this case persons with developmental disabilities.

A user-centered design is essential for the final result. In user-centered design the users are a part of the design process in one way or another. This can be done by e.g. letting the users test the product during the process. It can be hard for the designers to imagine every scenario and all the criteria that are important to the users. User-centered design helps making the product fit the users' needs. More advantages with user-centered design are that the product becomes more effective and safe and requires less redesigning. The disadvantages are that it takes time to plan and conduct tests with the users and that the product can get too specific in its use and may not be suitable for general use (Abrams, Maloney-Krichmar, Preece, 2004).

2.2 Research Methods

There are different types of research methods that can be used to gather information from users. It can be done by quantitative or qualitative research methods. The quantitative methods are used to gather numerical data such as the amount of errors and time spent on a task, whereas the qualitative methods are used to gather data about e.g. behaviors and opinions. In this project qualitative methods have been used, since it has been of importance to see how the user perceives the interface and there were only five users, which was not enough to gather statistical data.

There are many ways to gather information with qualitative methods. It can be done by interviews, questionnaires and observations. The method that is used in this project is based on observations. It enables a better understanding of the interactive interface and how the users receive it.

It is of importance to set up clear research questions and goals before each observation, to be able to provide useful data. The data from the observations, e.g. the reactions during the activity, can then be further analyzed. The gathering of data through observations can be done by e.g. video recording, photographs, data log and observer's notes (Preece, Rogers and Sharp, 2007, p356-359, 374). Video recording and observer's notes have been used in this project to gather information. Video recording is a good approach when working with users in a Snoezelen environment (Mertens, 2008, p.22), since they are not often able to express their needs and wishes verbally.

The benefits with video recording are that the session can be reviewed several times afterwards and it gives a detailed picture of the activity and problems with the design. The problem with video recording is that the view is limited. With notes only the important information, according to the observer's opinion, is noted. The observer's notes are then used in addition to video recording to get a complete picture (Preece, Rogers and Sharp, 2007, p295-297).

2.3 Bodystorming

Bodystorming is a way to broaden the perspective of how something is being approached or used, by participate in the interaction. To participate in the experience the designer assumes the role of the participants and imagines different scenarios. This is to gain better understanding and empathy with the participants (Smith, 2014, p.70).

Bodystorming can be performed in three different ways, design in place, prototype in place and embodied performance. Design in place is when a product or service is developed in the intended environment, to adapt it according to the surrounding. Prototype in place is when the product or service is tested in a simulated environment, which is a replication of the intended environment. In such environment, things in the surrounding can be changed to fit the conditions. Embodied performance was used in this project. Designers or other people, brought into the session, improvise as a participant to get the experience of the product or service (Schleicher, Jones and Kachur, 2010, p.47-48).

2.4 Risk Analysis

When making the interactive interface some safety measurements need to be considered. This is mainly to prevent users from getting injured and keeping the interface from getting damaged. By analyzing the different risks they can be reduced or avoided. When evaluating the risks, three things need to be considered according to Stanley Kaplan and B. John Garrick (1981): "What can happen? How likely is it that that will happen? If it does happen, what are the consequences?" The risks can then be predicted by setting up a risk analysis table, based on these questions. The risk factor of a certain scenario is the estimated probability and the estimated consequence. It can be defined as the product between the two factors. With this method, how the risk factor was calculated has to be considered when evaluating the risk factor. A high probability and a low consequence can give the same risk factor as a low probability and a high consequence, but the outcome is very different (Kaplan

and Garrick, 1981). The goal with the risk analysis is to lower the risk factor, and thereby the risks, during development.

2.5 Project Structure

While developing the interactive interface specific methods were used to get an overview of the structure of the project. This was done to simplify the work and make it easier to find errors.

2.5.1 Vertical and Horizontal Prototyping

When prototyping two different approaches can be used, horizontal and vertical prototyping, figure 2.1. Horizontal prototyping focuses on implementing many functions with few details, while vertical prototyping focuses on implementing fewer functions with more details. Vertical prototyping allows the user to test more functionality on fewer functions of the interface. Horizontal prototyping allows the user to test more functions but with less functionality. This is to see whether the functions work together or not. With the horizontal prototyping the user has to imagine how the details of the functions would work (Nielsen, 1993, p.94-95).

In this project vertical prototyping has been used, since the users in the Snoezelen institutions often do not have the ability to imagine different scenarios. That is why implementing fewer functions with more details has been of importance.

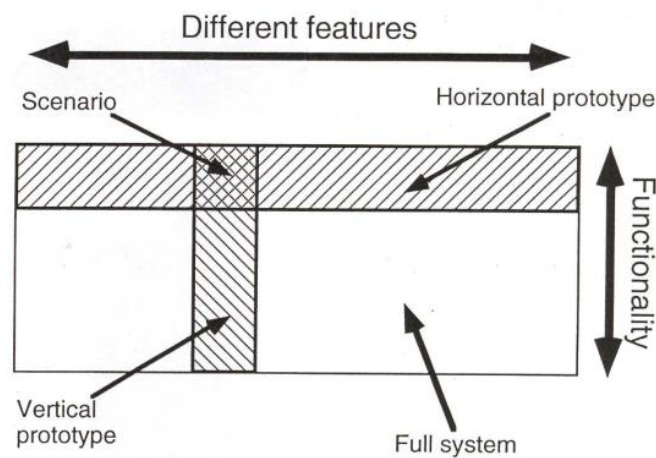


Figure 2.1 Shows the two types of prototyping. Horizontal prototyping has more functions (features) with fewer details (functionality). Vertical prototyping has fewer functions (features) with more details (functionality) (Nielsen, 1993).

2.5.2 Structured Design

Structured design was used to organize the design process. The basic concept of structured design is to define the different functions, problems or elements and the relation between them, to get an overview of the project. Dividing the design into

smaller subsections makes it easier to find problems in the design. Defining sections that are correlated makes it easier to distinguish pieces that might not be important to implement. In this way, the design process becomes more efficient. It will also ease the work for future developers if changes are required or unexpected errors emerge. One way of setting up a structured design is to use a flowchart to define the different functions or problems etc. (Yourdon and Constantine, 1979, p.15.18-21, 50). Figure 2.2-3, shows examples of structured design as a flowchart and shows the details of vertical or horizontal prototyping.

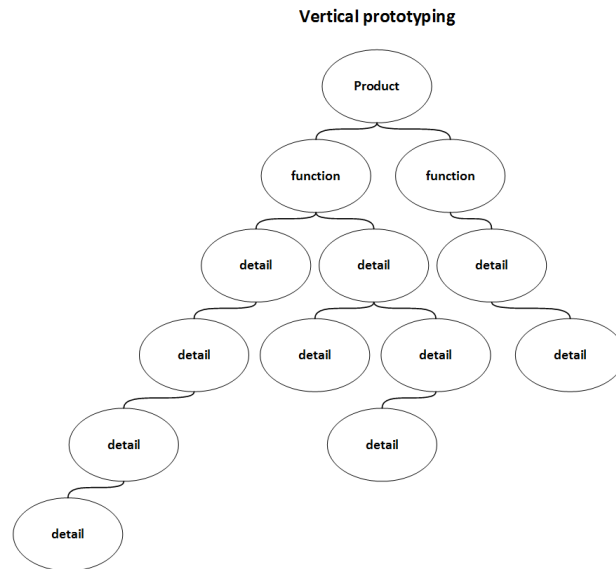


Figure 2.2 Structured design for a vertical prototyping shown in a flowchart. It has few functions but many details.

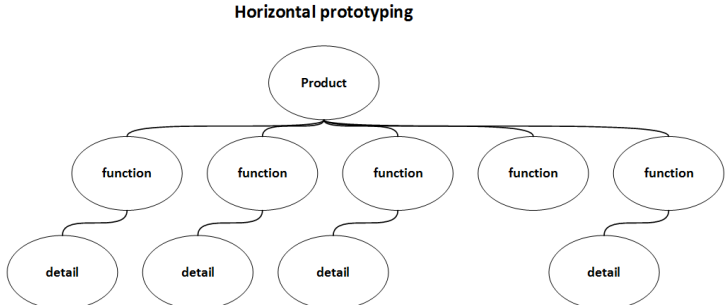


Figure 2.3 Structured design for a horizontal prototyping shown in a flowchart. It has many functions but few details.

2.5.3 State Machines

A state machine is a tool used during development to approach and solve problems. It can also be used to describe the solution for future developers. It is a graphic representation of the outputs and the inputs of a process. There are different kinds of

state machines, the one used in this project is based on the Moore machine. The outputs are represented as different states, and the inputs cause changes of states, figure 2.4. In the figure the circles symbolize states. A Moore state machine is always in one of its states, never in between. Depending on the input the state changes or stays the same. The arrows show how the state changes according to the input (Null and Lobur, 2006, p.135).

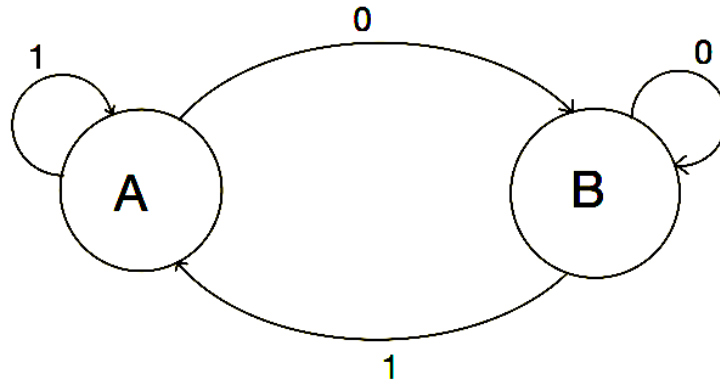


Figure 2.4 Moore machine. When input is 1 the state is A. When input is 0 the state is B

3 Pre-study

The definition of developmental disabilities is broad and encompasses many kinds of different cognitive, sensory and physical impairments, and this makes it difficult to implement a design in a way that appeals to a wide range of users. It is of importance to the design process to understand what the users are able to do and what they are not able to do (Preece, Rogers and Sharp, 2007, p.6). To make this possible the project was done in collaboration with mentors, staff at the Snoezelen institutions, and the users to meet their needs and wishes on how the interactive interface should work. The mentors have experience of interactive designs and the technical area, and the staff has experience of interpreting the users' actions. This chapter describes the Snoezelen institutions, and how they differ from each other. It provides an introduction to the users, who participated in this study, to better understand their abilities and disabilities. It also includes information about existing interactive Snoezelen equipment.

3.1 Snoezelen Institutions

The research for this project was done in collaboration with three different Snoezelen institutions, *Snoezelhuset* in Gentofte, Denmark, *Safirens Upplevelsehus* in Malmö, Sweden, and *Snoezelenverksamheten* in Lund, Sweden. Even though the rooms have similar themes the Snoezelen institutions have different approaches on how the environment will affect the user. The common factors in all three of the Snoezelen institutions are a waterbed with warm water, bubble tubes and ball pools, a white room and a dark room.

The feedback from the visits had a major influence on the decisions regarding the design. There have been four visits to each of the Snoezelen institutions. The first visit was an inspirational visit to get to know the staff and to get a better understanding of how the different Snoezelen institutions works and looks. To get an idea of what is commonly appreciated the staff explained what the world looks and feels like when having sensory deprivation. The users who were going to participate in the study were introduced on the second visit. At the third and the fourth visit the users tested the interactive interface. How it was approached and used was observed and recorded. The result will be further explained in chapter 5 and 6.

3.1.1 Snoezelhuset, Gentofte

In *Snoezelhuset* in Gentofte there are four Snoezelen rooms, det røde rum (the red room), det hvide rum (the white room), kuglerummet (the ball room) and hulerummet (the cave room). All of the rooms are small, and it is easy to add and remove equipments from the rooms. *Snoezelhuset* in Gentofte focuses more on the doezelen part. The users are mostly adults, and the staff adapted the practice and the environment to fit them the best. The rooms are more focused on how the activity will be in the different rooms, in other words, the room decides if the user should be aroused or calmed.

The red room, figure 3.1, is an active room, red arise the feeling of activity and excitement (Remdahl-Andrén and Malmgren, 2008). Having this first impression when entering the room allows the mood to change, to being awake and active, in other words, it is time to snuffle. In the room there is a red hollow bed made of wood with a subwoofer inside. When music starts to play the user can feel the vibrations from the subwoofer. The vibrations let the user experience their senses from both inside and outside the body.



Figure 3.1 The red room with bubble tubes (Gentofte Kommune, 2014).



Figure 3.2 The white room with a waterbed and an image from a liquid projector (Gentofte Kommune, 2014).

In the white room, figure 3.2, there is a waterbed and a big hammock, which stimulates the sense of balance. Besides stimulating the sense of balance, the room provides visually stimulation, with a liquid projector, and moving light on the wall and the ceiling.

In the ball room, figure 3.3, there is a box of plastic transparent balls. The room creates tranquility and lets the user relax.

The cave room, figure 3.4, is a darkroom with black lights. The room works as an escape from the reality where there are usually too many impressions that will activate the sense of sight. The room allows someone to relax, without having multiple things interfering to stimulate the same sense at the same time (Gentofte Kommune, 2014).

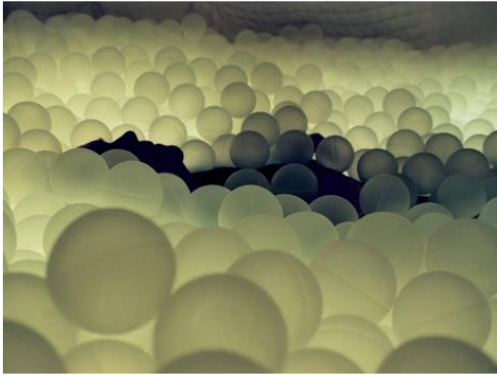


Figure 3.3 The ball room have a ball pool, with a warm light in the bottom to make the balls glow (Gentofte Kommune, 2014).



Figure 3.4 The cave room with black walls and black interior. Strands shines in different colors (Gentofte Kommune, 2014).

3.1.2 Safirens Upplevelsehus, Malmö

Safirens Upplevelsehus (experience house) is an institution in Malmö that work with Snoezelen environments. There are four rooms, called Solrummet (the sun-room), Månrummet (the moon-room), Vindrummet (the wind-room) and Himmelriket (the kingdom of Heaven). The rooms at *Safirens Upplevelsehus* in Malmö are more user adaptable, which means that the room can get more arousing or more calming dependent on the need of the user.

The sun-room, figure 3.5, is an active room, the color theme is red and yellow to encourage activity. The room is big and has a big ball-pool. The plastic balls are red and yellow, as the color theme of the room.



Figure 3.5 The sun room with a big ball pool.

The moon-room is a dark room with black lights. Some equipment is interactive, and by sound triggers the lights will turn on and off.

In the kingdom of Heaven there is an interactive interface from the SID project, an adapted waterbed. It generates soundscapes from the movement of the water that is created by the user (SID, 2014). In the ceiling there is a mirror ball that reflects the light in different directions and creates moving lights.

In the wind room, figure 3.6, the colors are mostly blue and purple. The blue color is meant to instill the feeling of calm and serenity, which also can lead to being more focused and productive (Remdahl-Andrén and Malmgren, 2008). This is good quality to help the user to feel confident to explore. In the room there are fans, and lightweight fabrics hanging down from the ceiling. The user gets to experience the feeling of the wind when the fans blow. The fabrics contribute to the visual effect of the wind, when flying around.



Figure 3.6 The wind room with fans and lightweight fabrics.

3.1.3 Hjälpmedelsverksamheten, Lund

The Snoezelen institution in Lund has two Snoezelen rooms, vita rummet (the white room) and mörka rummet (the dark room). The rooms are big and to add and remove equipment in the rooms are difficult.

In the white room, figure 3.7, there is a waterbed, and a small rocking bed to stimulate the sense of balance. There is also a liquid projector and bubble tubes to create visual effects.

In the dark room, figure 3.8, they have a ball pool, with a light source in the bottom to lighten up the balls. The rooms have strands in different colors that shine when exposed to black lights.



Figure 3.7 The white room with a small rocking bed and mini-light strands (SID, 2014).



Figure 3.8 The dark room with a ball pool and strands that light up from the black light (SID, 2014).

In addition to the Snoezelen institution, the collaboration in Lund has been with the staff working at *DUMLE* (Du Utvecklas Med LEk, translated you develop by playing), which is a part of the habilitation center (Hjälpmedelsverksamheten) in Lund. At *DUMLE* the focus is to arouse the childrens' curiousness to explore things they usually are not able to experience. There are different kinds of equipment and toys that are rebuilt in a way that makes them more manageable and understandable for children with disabilities. At *DUMLE*, parents who have children with cognitive disabilities can get inspired and get more knowledge on how to rebuild toys and what kind of equipment that attracts their child. *DUMLE* lets these parents bring their children to try these things out before buying it for themselves.

3.2 Participants

The target group involves adults with different grades of disabilities. Impaired vision has been a common factor, though the degree of impairment has been different for each user. The staff at the Snoezelen institutions selected the individuals who participated as users. Selection was done with regards to the individual's regularly abilities, disabilities and interest. Five different users who visit the Snoezelen institutions have participated in this study. In this report, all the participants have been given fictitious names.

The user at *Safirens Upplevelsehus* in Malmö is called Bane. He has impaired vision, doesn't have the ability to talk and uses a wheelchair. Although he cannot talk, he expresses satisfaction or dissatisfaction through vocalizing sounds. He also has a habit of biting his arms, which is an expression of engagement, since he cannot use words. It's a way for him to release tensions from strong impressions, both good and bad. By showing him interest and distract him with other things, its behavior can be reduced. Despite his poor vision, he can notice very bright light, which is close to his face. He can move his arms and grab things with his hands, but needs to have the arms close to his body. This makes small things more attractive to him. Small things are also easier to grab and interact with, in a way that he can decide whether he wants to have it close to his face or not. Bane gets much focused and vocalizes sounds when he is comfortable with a situation or when he gets interested.

At *Snoezelhuset* in Gentofte, there were four users with different grades of disability, Ivy, Selina, Robin and Alfred. Ivy is most developmentally advanced of them all and loves to talk in her own way, which is understandable for the staff, whom has been spending time with her. For others who haven't met her before it might be a bit hard to understand. Her ability to talk makes it easier to understand, with some help from the staff, if she likes something or not. Ivy can move her arms and her hand. Her vision is impaired, but she can see lights and structures. Selina uses a wheelchair and can move her head and her hands, though she can only move her hands close to the body. She can spread her fingers, but not use them to grab things. Her way of showing satisfaction is to spread her fingers. Robin can move his arms, hand and fingers. He is very curious, he likes to use his fingers to fiddle with things and explore. He can walk, and walks away if the situations are not appealing to him. Alfred is blind and uses a wheelchair. He can move his arms and hands and likes to have his hand close to the neck. He uses his fingers to explore. To have things close to the neck is an advantage in his case.

In Lund, there were no users involved in the project. Instead the feedback came from the staff, who has many years' experience of working with children with cognitive disabilities. The staff provided feedback on whether the interaction between the user and the interactive interface would be beneficial for the user or not. The collaboration also provided general information and advices to think about while developing the interactive interface.

3.3 Interactive Snoezelen Equipment

Even though it's not common, there is equipment in Snoezelen environments that are interactive. Some are interactive by push buttons or sound triggers. In this way the users can interact with the equipment by pressing a button or clap their hands, and thereby get a response. An example of existing interactive Snoezelen equipment with "push buttons" is interactive stepping stones, figure 3.9. The user uses the body to climb and press the stones to activate e.g. lights or bubble tubes (ROMPA, 2012).



Figure 3.9 *Interactive stepping stones. The user presses the stones to activate lights (ROMPA, 2012).*

Other equipment uses close interaction, which makes it easier for the user to access and understand. With close interaction the user gets a direct response from the equipment, compared to the push buttons where the response is given at a distance. The interactive design in the SID project uses close interaction. Examples of designs in the SID project are LivelyButton, HugBag, ActiveCurtain and MalleablePillow. The LivelyButton, figure 3.10, is a wooden box with rotating metal springs and fabric on the top. When the user approach or touches it, the color shifts and the box vibrates. The more proximal is the user, the faster it rotates and shifts color.

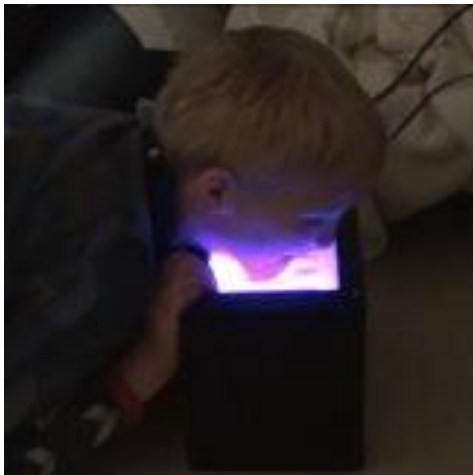


Figure 3.10 LivelyButton lights up and vibrates when approached or touched (SID, 2014)



Figure 3.11 HugBag changes color and makes sounds when the user hugs it (SID, 2014).



Figure 3.12 ActiveCurtain creates a new image when pushed (SID, 2014).



Figure 3.13 MalleablePillow lights up in different colors when moving the marbles (SID, 2014).

3 Pre-study

HugBag, figure 3.11, is a big soft transparent ball fixed to a plastic base. When hugged the ball changes color and makes sounds depending on the location and strength of the hug. The ActiveCurtain, figure 3.12, is a fixed and elastic curtain that is lightened up in a color spectrum. When the user pushes the curtain the image changes depending on the place and the strength of the push. MalleablePillow, figure 3.13, is a pillow filled with marbles and fabric that rustle. When touched and knead, the pillow emits light, the more it's kneaded, the more intensive is the light (SID, 2014).

4 Proposed Solutions

After the inspiration visits, many different ideas of design emerged. The idea suggestions are described in this chapter. Two of the ideas were further developed. One idea was to make already existing equipment in the Snoezelen environment, the bubble tube, interactive. The other idea was more of a challenge, when trying to figure out a new interface that hopefully would be popular, but it might also be a total flop. This other idea Glownado, is an inspiration of several equipment in the Snoezelen environments.

4.1 Brainstorming

There were many ideas in the beginning, after the inspirational visit.

One was to modify a kid toy, were figures pop up when pressing buttons. The idea came from a box with push button that are used in one of the Snoezelen Institutions. This could benefit the user when getting a direct response from their actions, in this case by pressing buttons. Having something that pops up will also provide a surprise element.

In the wind room, at *Safiren*, they had big fans standing on the floor. They were controlled by push buttons, to turn on and off the fans and to set the fan speed. One idea was to control these fans with Kinect sensors, to make them more interactive. The wind would change in intensity depending on how the person moves in the room.

In the dark room, at *Safiren*, they have a box on the wall with feathers inside, figure 4.1. A fan is put inside the box and the feathers provide a visual effect when flying around. An on/off button controls the fan. The idea was to control the box with the body, where the fan activates when a person touches the window of the box.

Another idea was to stimulate the sense of balance using the idea of a baby rocking chair. When put into motion the baby rocking chair would make sounds and change colors.

Next idea played a lot with the sense of touch. It was based on a box, were “grass” were supposed to grow when touching the surface. The grass will then provide a smooth tactile feeling.

Another idea was to implement sound in fiber optics that activates when touched. To place balls along the fiber optics was also an idea, to create a different tactile feeling.



Figure 4.1 The user can switch colors with the buttons, and turn on and off the fan. The feathers inside gives a visual effect of the wind, the fan is creating.

4.2 Interactive Bubble Tube

The bubble tube, figure 4.2, is common equipment in all Snoezelen places. It is one of the first things that are recommended to use when designing a Snoezelen room (Mertens, 2008, p.50). The bubble tube is a tube filled with water, with bubbles generated by an air pump in the bottom. Controlling the air pump sets the amount of bubbles. The tube has a light source that can be set to the desired color. It stimulates the user's vision, makes them concentrate and stimulates a tactile feeling when the tube vibrates. The staff or carer handles the settings of the bubble tube. The users can have access to push buttons, where they can turn off or on the bubbles and change the color of light. It is only the users who have the ability to use push button that can benefit from this.

The initial idea was to make the bubble tube interactive and more accessible for a larger set of users by removing an actual push button and instead make their bodies function as a button. The interactive bubble tube gives the opportunity for the user to explore with their whole body or just a few body parts depending on the user. The help of a distance sensor and a capacitive sensor would do this. The distance sensor would detect the distance between the user and the bubble tube. When the user gets closer to the bubble tube the intensity of the bubbles will increase. A capacitive binary sensor that can detect touch, using 12 different electrodes, would be added around the tube, see figure 4.3. Each electrode controls one of the colors, red, green and blue. Depending on how many of the electrodes that are active the intensity and the color of the light will change. The staff or the carer would use a control box to adjust the settings to fit the current user. The amount of capacitive sensors, which needs to be activated for increasing the light, is set with the control box. The control box also sets the threshold for the distance sensor, for when it should detect an approach. In order to contribute with tactile properties a layer of various transparent

materials would be added around the tube to arouse the curiosity of exploring the whole tube and thereby activate more electrodes.

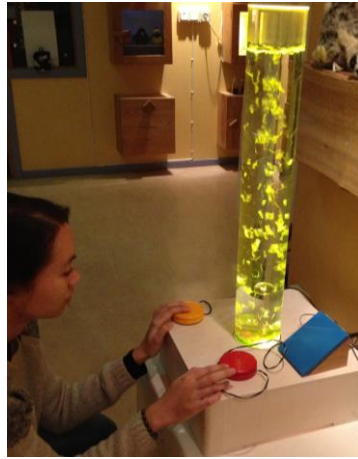


Figure 4.2 The user can change the color of the bubble tube by pressing the buttons.

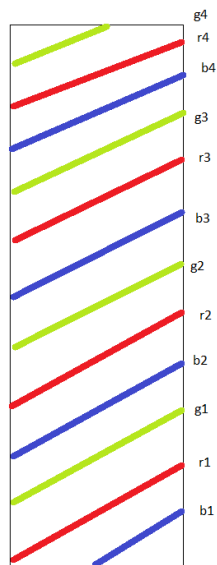


Figure 4.3 A sketch of the interactive bubble tube and how the 12 electrodes should be placed around it. Each electrode controls one of the colors, red, green and blue.

4.3 Glownado

The word Glownado is composed by the words glow and tornado. The main inspiration came from the wind room at *Safiren* in Malmö. Here they used wind as a tactile element as well as the more “traditional equipment”, e.g. fabrics with different texture. Another inspirational source was Paletto, figure 4.4, which is a box that is placed on the wall. It has different buttons that make sounds when pressed, and the surface of the buttons have different tactile surfaces and colors. Bane was also a major inspiration. He held a color changing egg, figure 4.5, in his hand and moved it back and forward close to his face to experience a visual effect. It made him calm and focused. By combining these inspirations, an interactive “wind box”, Glownado, was going to be developed, where a fan activates when touching different objects.



Figure 4.4 Paletto makes different sounds when the user presses the buttons (ROMPA, 2014).



Figure 4.5 A color changing egg changes color slowly (ROMPA, 2014).

There were many different ideas of how the “wind box” should look like and which sensors would fit the purpose the best. The overall idea was to make the design small and have a fan as the main attraction. By making it small it would be accessible for a larger set of users, since it can be placed where they would benefit from it the most. An interactive fan has never been implemented in Snoezelen equipment before. Most of the fans used in Snoezelen are controlled by the staff who can turn on and off the fans and regulate the fan speed. It would therefore be interesting to observe how the users would receive the surprise element the wind creates, if the wind would keep their interest to explore. The users would gradually control the intensity of the fan. A light source would be used to attract the user to interact with the interface. To stimulate the sense of hearing, Glownado would provide a sound when set in motion. The final version of Glownado and the design process will be explained in chapter 5.

4.4 Concept selection

Only one of the designs was implemented. The reason was that there was not enough time to do both. To develop existing equipment, the bubble tube, was a “safe” idea

knowing that it is already popular. The idea was to make it more accessible, but it would still be hard to come close to the tube when sitting in a wheelchair. Glownado was more desirable, since it was completely different from the existing equipment. The possibility to make it small also made it more desirable, since it can be adapted to the user's situation.

5 Design Implementation

When designing the interactive interface, it is important to have in mind that the user should shift between to snuffle and to doze. Meaning that the user is in a stage when neither understimulated nor overstimulated but something in between. When understimulated there is lack of stimulation to activate the senses. If overstimulated there might be too much stimulation of the senses at the same time, which makes the user no longer susceptible to the information that is relevant and thereby misses the purpose of the interactive interface.

The design has been developed using an iterative design process. Different iterations have been made according to the observations, as well as the opinions of the mentors and the staff, to fit the user's needs.

Small steps have been taken and after each implementation Glownado has been improved based on the usage. The three outputs (wind, light, and sound) have been tried out separately, and sequentially been added to see whether they work together and complement each other or not. In other words, in this project the wind feedback, which is the main interaction element, was the first one to be implemented. The light was then implemented, and its behavior was set to support the fan's behavior. Having these two outputs set, sound was added. The behavior of the sound was also implemented to complement the fan. For more information about how each of the components works, see Appendix A.

During each iteration bodystorming was used to try to experience how Glownado was going to be approached by the users by assuming different scenarios. While bodystorming, elbows, hands and head were used to approach the interactive interface, since it was pointed out, by the staff, that many users preferred to interact this way. As a part of the design process, the users' interactions were observed in the Snoezelen environment. The development continued based on these observations. Before introducing the interactive interface to the users, a risk analysis was made to evaluate and prevent risks. The risk analysis table for each visit can be seen in appendix B.2.

5.1 Overview, Criteria and Structured Design

The ideas of how some part of the interface should look like were decided after the inspirational visits. Different criteria were set when developing Glownado. The main criterion was to implement a gradual response, to make the users understand that their actions are the cause of the responses given by the interactive interface. It should also respond immediately to make the user understand that the interactive interface is responding to a specific action. Another criterion was to implement a fan in the

design, to work as a tactile element, since it has not been used as an interactive element in Snoezelen environments before. The wind would also work as an element of surprise. Implementing sound and light were set as criteria to provide a multisensory stimulation, which will complement the tactile element from the wind. Light and sound are more commonly used to stimulate senses in Snoezelen environments, so they will not be used as the main element in Glownado. Another criterion is to develop an interactive interface where different body parts can be used to interact with it. Vibration was also desirable, but this was not set as a criterion. Another desirable feature was to stimulate the sense of smell. It is rare, having something in the Snoezelen environment that activated the sense of smell, because it is hard to control, since it lingers. With a fragrance in Glownado, the fan would help spread the essence. Therefore, the idea of implementing a fragrance in the design was taken into consideration.

After the first visit with the users more criteria were set. The first, was to make the design small, to make it easier to grab, since some of the users had this ability. It can also easily be manipulated with the back of the hand, which would suite the users who prefer approaching things in this way. By having a small design the user gets to decide by their own if it is more comfortable having it close to the body or not. This gives the impression of having control, which is the same as saying, having the control of something gives one confidence in exploring new things. According to Lone at *Safiren* in Malmö, it is beneficial to have mobile equipment, because it makes it easier for the user to notice it. To create motion in Glownado was set as a criterion. The user should also be able to put the design in motion by a small effort, to make it more suitable to a wide range of users, where some are stronger while others are weaker and more careful. Another criterion was to make the light as strong as possible to make it more noticeable for users with partially impaired vision. The last criterion was to make a robust and sturdy design, to fit the users with impaired vision. Making Glownado robust and sturdy can also add to the feeling of security and confidence. Giving an example, when feeling dizzy it feels safer to lean on something that doesn't give way. When having vision impairment it feels safer touching something robust and sturdy than touching something that is soft and unstable. Having all these criteria set after the two visits the different senses, sight, hearing, smell and touch would be stimulated when using Glownado.

In this project an Arduino board was used. It consists of 14 digital input/output pins, six of them can be used as analog output pins, using PWM (pulse width modulation), and six analog input pins. To program the microcontroller, the Arduino IDE (Integrated Development Environment) is used (Arduino, 2014). The Arduino programming language makes it simpler for the designer to program. To give a response to the users' actions different sensors are connected to the input pins and the Arduino can control different actuators on the output pins. The sensors were chosen to be able to give a gradual response to the actuators, the outputs. With the sensors, the interactive interface can detect the actions of the user and give a response through the outputs the interface generates, such as light, wind and sound.

The schematics and the code of the final version of Glownado can be seen in appendix C.

A flowchart of the structured design shows how the work would develop, figure, 5.1.

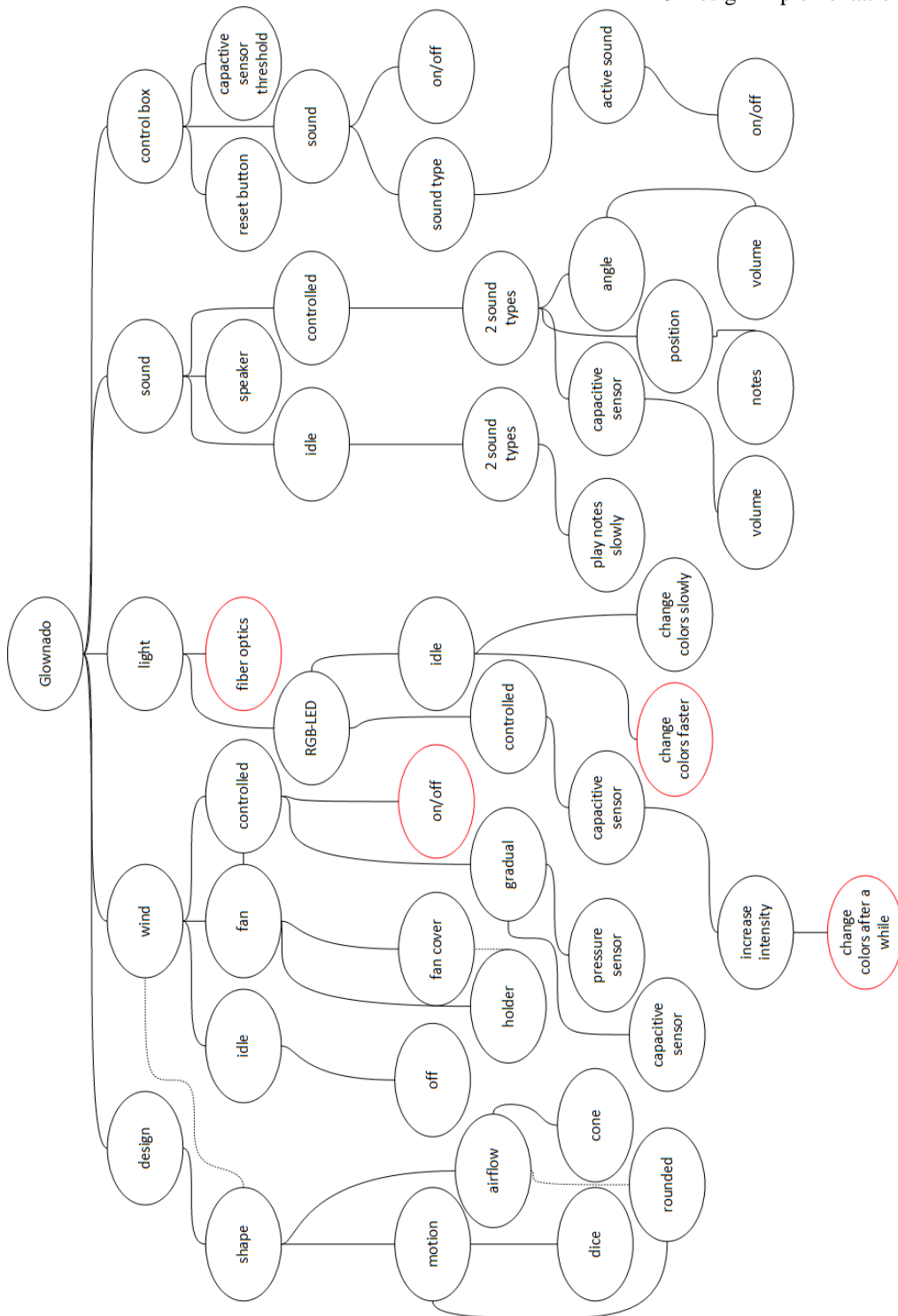


Figure 5.1 The structured design for Glownado. The red parts were never implemented, since they turn out to be irrelevant to the design.

5.2 Physical Material Development

Different fans were tested to find the one most suitable for this project. The comparison is shown in appendix B.1. Having a fan with strong airflow was prioritized, since it will be reduced when placed in the design. It also needed to be enough powerful to notice the gradual changes. To be able to control the fan gradually with Arduino, a PWM fan was desired. A small fan with low noise level was a desirable feature. The fan that was chosen for Glownado was a Gentle Typhoon. The powerful fan resolves in a higher noise level. Since no other fans tested had such good airflow, the noise level could be overseen.

5.2.1 Implementing the Design

The first idea was to build the interactive interface as a dice, figure 5.2. The shape would make it easier to grab, and quite easy to flip over. The size of the dice would be approximately 13*13*13 cm, to be able to fit the fan in the design. What needed to be taken into consideration was the fact that the fan only would blow air in one direction. The fan would not make sense if it were not directed towards the user. With this design the fan could end up upside down. The airflow would also be significant reduced in this design, since it would have a shape of a cube and therefore won't direct the airflow in the best way. All the sides would also be covered, with only small holes on the sides to let air in and out. The cube could be made bigger, but then the whole interactive interface would get too big. This would not fulfill the criterion to create a small interactive interface.



Figure 5.2 The first idea was to shape Glownado as a dice.

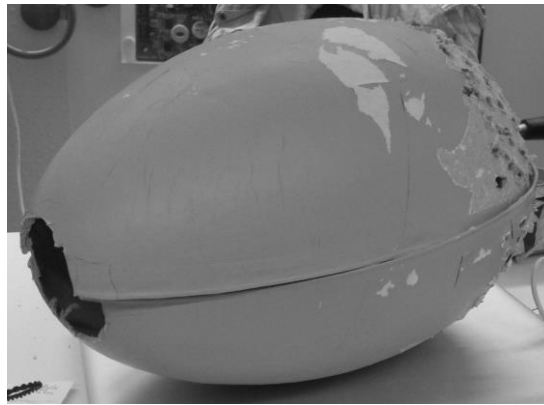


Figure 5.3 A prototype using an egg-shaped cardboard was investigated. Holes were made in the bottom to allow air into the fan and a bigger hole was made in the top to let the air out.

The inspiration for the next idea came from a hairdryer, which is shaped like a cone to direct the airflow. Different objects and materials shaped as cones were tried out, for example plastic coffee filter, and oven paper. This was done to determine which shape produced the best airflow. The conclusions were to place the fan in a cone, and having the fan close to the sides of the cone. Otherwise the air would escape on the sides between the walls of the cone and the fan, instead of going through the fan. A

5 Design Implementation

box would be placed beneath the fan. To allow air to the fan, the box should have holes on the sides. By having a rounded bottom, the design could be set in motion. It would also clearly show how the design should be placed, to always have the fan directed upwards, i.e. in the direction of the user. To get the feeling of how rounded the bottom should be, to be enough sensitive when being pushed or touched, the first prototype was built using a big egg-shaped cardboard, figure 5.3.

Second prototype was built with a sheet of metal as the bottom and a jug as the top. Several holes were drilled in the metal and then it was shaped as a cylinder. The bottom of the jug was cut off and used upside down, this can be seen in figure 5.4. This prototype was built to examine if the airflow needed to emerge into a smaller opening, to focus the air, or if the small gradient of the jug was enough.



Figure 5.4 Prototype to examine the gradient of the cone.

For further development of the design, the inspiration came from a Roly-poly toy. The toy is rounded and heavy at the bottom and can never tip over, it always bounces back, figure 5.5.

A sketch was made for the design of Glownado, figure 5.6. The lower part is rounded with evenly spaced holes all around, to create an airstream to the fan. The Arduino would be placed in the lower part, and the fan would be placed in the upper part. The upper part is shaped as a cone to get the wind more concentrated and directed. To draw attention to the fan a light would be placed in the upper part. To prevent Glownado from tipping over, a ring is placed around the lower part of the design.



Figure 5.5 The toy, Roly-poly, can never tip over when pushed (Pink and Blue, 2014).

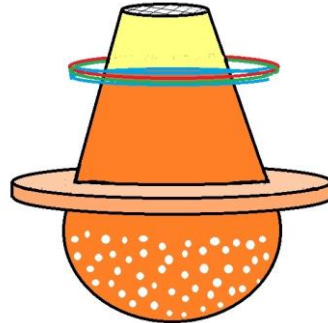


Figure 5.6 Sketch of Glownado. The yellow, and rings of red green and blue shows where the light should be placed. The ring prevents Glownado from tipping over.

From the sketch a new prototype was constructed, figure 5.7. The lower part was a plastic lampshade, where holes were made. The upper part was the same as before, when using the lemonade jug. The size of the holes in the lower part was examined to see whether they would provide enough air to the fan. The lampshade was quite unstable due to the holes and the soft plastic it was built of.



Figure 5.7 A prototype made from the sketch. Used to examine the size of the holes to provide a good airflow to the fan.

To get a desirable shape and size, a 3D printer was used to create a shell for the interactive interface. 3D printer creates a three-dimensional solid object from a digital model. The software application used was AutoCAD. The CAD model of the design was constructed in two parts, one upper part and one lower part. The lower part is a

5 Design Implementation

cylinder with a rounded bottom. The color black was chosen for the lower part. While the upper part was made in a transparent white, making the light more noticeable. This design choice was made since the fan is the main attraction and the focus should, therefore, be on the upper part. The size of the design was set to fit the fan, the Arduino and components, figure 5.8. The shape of the bottom was a compromise between having it completely rounded, making it more sensitive to a weak push, and to have it flatter, which would make it more stable. The decision was to make the bottom quite flat, and if noticed that the bottom should be even more rounded this could be changed afterwards. The final printed version is shown in figure 5.9. Since the plastic, which Glowmodo is made of is light, and the placement of the fan makes the top of the design heavy, a piece of metal was placed in the bottom of the design to make it more stable. To compensate rough pushes three water hoses, formed as rings, were added around the lower part. The upper part and the lower part were connected to each other with screws.

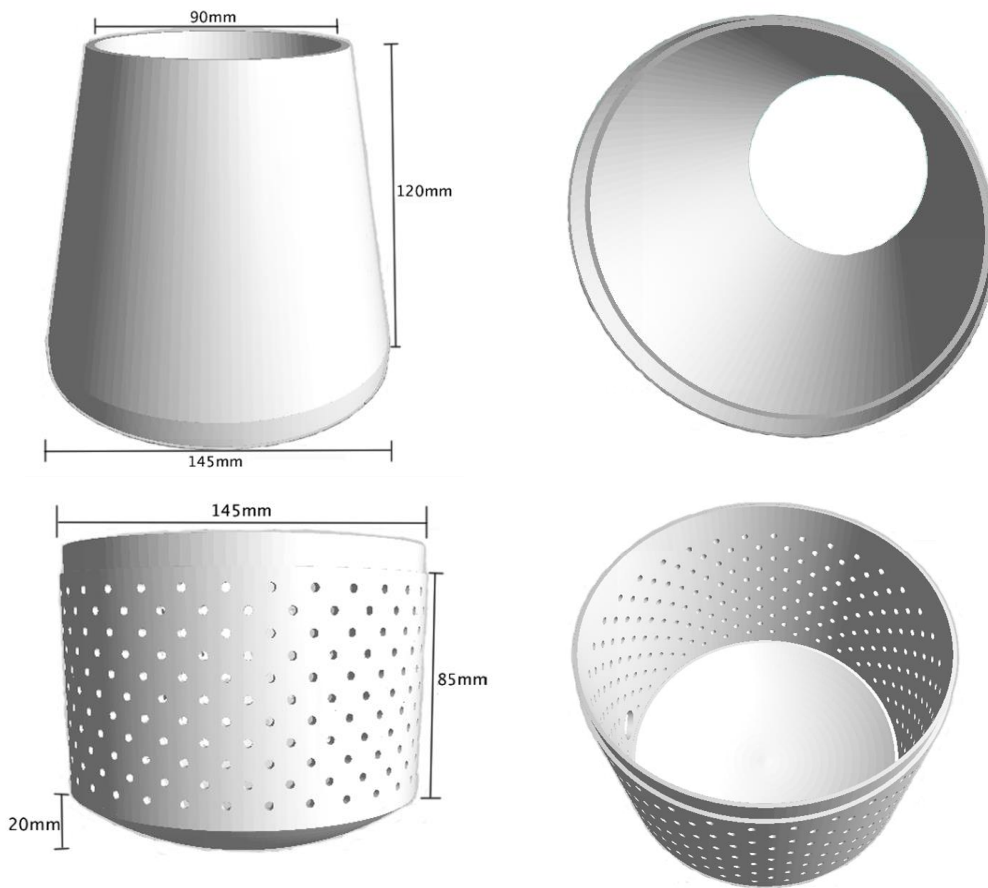


Figure 5.8 Upper left figure shows the dimension of the upper part. Upper right figure shows the inside of the upper part. Lower left figure shows the dimension of the lower part. Lower right figure shows the inside of the lower part. The bigger hole to the left is used for the cables.



Figure 5.9 The final 3D printed version. The lower part is black to avoid attraction and the upper part is transparent white to attract the user to the wind.

5.2.2 Implementing Wind

Wind was used as a tactile element. Different types of fans were tested in this project, five CPU cooling fans and a hair dryer fan motor, appendix B.1. Most of the cooling fans supports PWM, (Pulse-Width Modulation). It was desirable to use a PWM fan, since it makes it easier to control the speed of the fan gradually through the Arduino's PWM output pins. The airflow and the size of the fan were also important. A powerful airflow would be more noticeable than a weaker airflow. By having a small fan the interactive interface can be made smaller.

At first, the fan was attached to the top of the design with a lampshade holder. The frame of the fan was cut off. It was then attached to the lampshade holder, which had been bent to fit the design, figure 5.10. Screws were used to secure the lampshade and the fan to the design, figure 5.11. For the safety of the user, a fan cover was added to the top to protect fingers and other things from accidentally fall into the fan. In this solution the screws had sharp edges. To mitigate this, the lampshade holder was removed and a second fan cover was added below the fan. The fan cover acted as a holder as well as a protection for the cables below, figure 5.12.

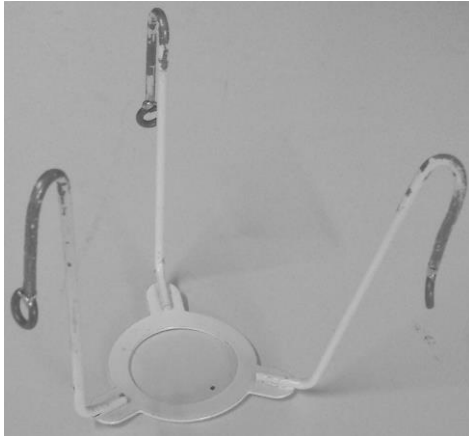


Figure 5.10 The lampshade holder. The fan is attached to the round part.



Figure 5.11 The lampshade holder and the fan cover are attached to the design by screws.



Figure 5.12 Seen from the inside. The fan is attached to a fan cover at the bottom of the upper part of the design.

To control the fan gradually, the first idea was to use pressure sensors. The sensors were supposed to be placed around the upper part of the design and a layer of foam rubber should be placed over the pressure sensors. The foam rubber would create a gradual feeling when pressing. Since the design is supposed to move when being touched, it would be hard to notice the gradual feeling from the pressure, instead it would create a feeling that the response is given by the motion. The foam rubber would also dim the light, which led to the disposal of the idea. The solution was to control the fan with a capacitive sensor, which reacts to the difference in electricity when e.g. being touched. The change in electricity then occurs due to the electrical capacitance of the human body. It would detect the distance between the user and Glownado. A closer approach would resolve in a higher fan speed and thereby more

intense wind. The following equations show the relation between the fan speed and the value of the capacitive sensor.

The fan speed is given by the equation 5.1 if *capsense value* ≤ *maxCapSense value*

$$fanSpeed = \frac{(capsenseValue - threshold) * (fanSpeed_{max} - fanSpeed_{min})}{capsenseValue_{max} - threshold} + fanSpeed_{max}$$

[Equation 5.1]

The fan speed is given by equation 5.2 if *capsense value* > *maxCapSense value*

$$fanSpeed = maxFanSpeed$$

[Equation 5.2]

<i>capsenseValue</i>	Value of the capacitive sensor.
<i>fanSpeed_{max}</i>	Maximum fan speed. The value used in the code was 255.
<i>fanSpeed_{min}</i>	Minimum fan speed. The value used in the code was 60.
<i>capsenseValue_{max}</i>	Value of the capacitive sensor when the distance between the user and the design is 0. Value used in the code was 1500.
<i>threshold</i>	Value for when an approach should be detected. Value used was 50-700.

The constructed capacitive sensor uses a RC circuit, which is a resistor in series with a capacitor, figure 5.13, to detect a touch or an approach. The sensor can feel through plastic, wood, ceramic and other isolated material with thickness of more than 6 millimeters. The sensor that was used detects a distance of approximately 6 centimeters. As a sensing material, a conductive fabric was attached to the top of the design, figure 5.14.

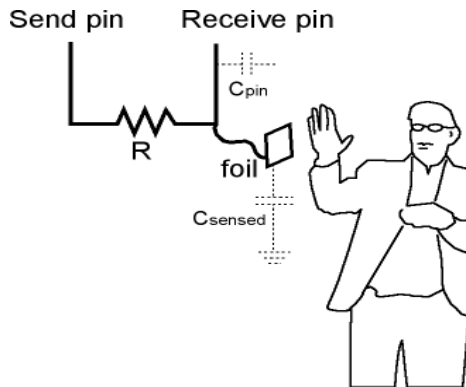


Figure 5.13 Schematic of the capacitive sensor. A resistor is connected between the sensor pin (send pin) and the receive pin. A capacitor (*C_{sense}*) in parallel with the human body and a floating capacitor (*C_{pin}*) stabilize the value (Arduino, 2014).



Figure 5.14 Conductive thread was sewn in a spiral and seven stripes of brown conductive fabric were added to work as a sensing material.

A white silk fabric was sewn to dress the top of the design. Conductive thread was sewn in a spiral shape around the fabric. Seven stripes of a brown conductive fabric were added vertically to the silk fabric. By adding them vertically, more fabric will be available to grab. It also allows more light through the design, than if they were placed horizontally. The conductive thread works as a connector to the conductive fabric as well as sensor. Both the capacitive thread and fabric also works as a tactile element to Glownado. Velcro tape was added to the silk fabric and to the 3D-printed upper part to keep the fabric in place.

5.2.3 Implementing Light

An IKEA RGB-LED was used as a light source in this project. By combining red, green and blue color LEDs in different intensities, a spectrum of colors can be created. A driver was used to supply the RGB-LED with 12 V.

The RGB-LED was placed on top of the fan, to work as a light source. The light was placed to be visible only at the top of Glownado. This is to draw attention to the top half of the design, which contains the capacitive sensor that activates the fan to give a wind response. The white silk fabric that was used for attachment of the capacitive sensor was chosen to reduce as less light as possible. It also creates a contrast between the white fabric and the dark brown conductive fabric. This clear contrast provides a visual effect.

5.2.4 Implementing Sound

Sound was used to stimulate the sense of hearing. In this project a musical instrument shield for Arduino and a small active speaker was used to produce the sound. It was chosen since it can create a variety of sounds through different instrument banks and notes, using the MIDI sound protocol. One can choose from two instrument banks with respectively 128 and 60 numbers of different instruments, such as piano, woodwinds, saxophone and percussion sounds. With some of the instruments it is possible to play melodies, by selecting the note and volume. An active speaker needs to be connected to the music shield to play the sounds (Sparkfun, 2014).

The speaker was placed in the lower part of the design, to give the impression that the sound comes from Glownado, figure 5.15. The speaker partly blocked the flow of air to the fan, but it did not significantly reduce the airflow out of Glownado. The speaker has internal batteries and needs to be turned on manually, though a button on the speaker, before a session and turned off manually after the session.



Figure 5.15 *The speaker, pink, is placed inside of Glownado.*

An accelerometer was used to control the behavior of the sound, because of its ability to measure movement and tilting of the design. These features couldn't be found in the other sensors, such as pressure sensor, ultrasonic sensor, capacitive binary sensor and Kinect, that was tested during the development of Glownado and bubble tube. Those were, therefore, not used in the final version of Glownado. The accelerometer being used in this project is a triple axis accelerometer, ADXL335. It has three analog outputs for the x-, y- and z-axis and senses the orientation in relation to the gravity. The accelerometer was used for measuring tilt, by using pitch and roll angles. This can be illustrated as the nose of an airplane going up and down and the roll is the rotation around the axis that goes from the nose to the tail. The tilting magnitude of the accelerometer is measured by calculating the hypotenuse of pitch and roll angles, while the direction is measured by calculating the pitch and roll angles. Figure 5.16 illustrates the tilting magnitude and the tilt direction.

To more accurately measure the tilt angle, the accelerometer was placed in the center on top of the fan. It will measure the same value when having Glownado in the same tilting magnitude but in different tilt direction. Having it on top of the fan, the values were unstable because of the vibrations the fan introduced by dynamic acceleration the fan is activated. To stabilize the values of the accelerometer a low-pass filter was implemented. The low-pass filter smoothens the signal and attenuates high frequency values, introduced by the activated fan, equation 5.3.

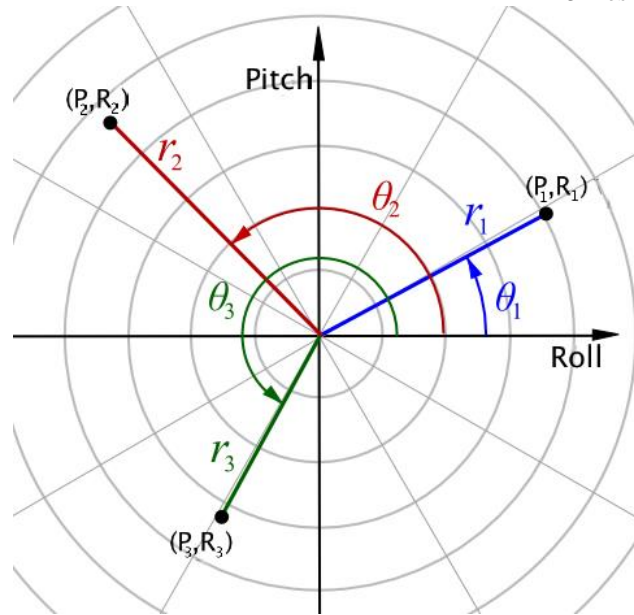


Figure 5.16 The figure shows the tilting magnitude (r) and tilt direction (θ) in a two-dimensional coordinate system. P is the pitch angle and R is the roll angle. (Wikipedia, 2013).

$$\text{current value} * \text{weight} + \text{previous value} * (1 - \text{weight}) \quad [\text{Equation 5.3}]$$

current value
previous value
weight

Current value given by the accelerometer.
Previous value given by the accelerometer.
The sensitivity of the low-pass filter, a lower value gives a slower filter but more stable values. The weight used was 0.3

When Glownado is set in motion the accelerometer is being activated, and different notes are played at different volumes.

5.2.5 Implementing Control Box

Implementing a control box makes it possible for the staff to change the settings to fit the current user, this will also broaden the range of users.

Electric potential from the human body differs between different persons. This can depend on what the person is wearing or if the person is using a wheelchair, etc. To compensate this, the values of the capacitive sensor need to be controlled to fit the current situation. A potentiometer was used to enable the possibility to change the sensitivity of the capacitive sensor. It allows the staff or the carer to change the threshold of the capacitive sensor, which determine when the fan should start to activate.

An external reset button was implemented, to reset the Arduino. This could be used if the capacitive sensor would behave different when the circumstances have changed.

To control the sound, a switch was added to the control box to be able to turn the volume on and off, if the user does not desire sound. This can occur if the user does not find the sound pleasant, or if it doesn't have any affect, for example, when the user is deaf or if it creates too much impression on the same time, then it might be good to be able to turn the sound off to spare the surrounding people if they don't find the sound pleasant.

5.3 Interface Behavior, part 1

In a truly interactive environment, in different situations there can be different responses to various actions. In the same way, Glownado should react in different ways depending on how the user interacts with it. Implementing this, different modes were set to control the behavior of Glownado. In the first implementation the modes were the *sleep mode* and the *active mode*, figure 5.16.

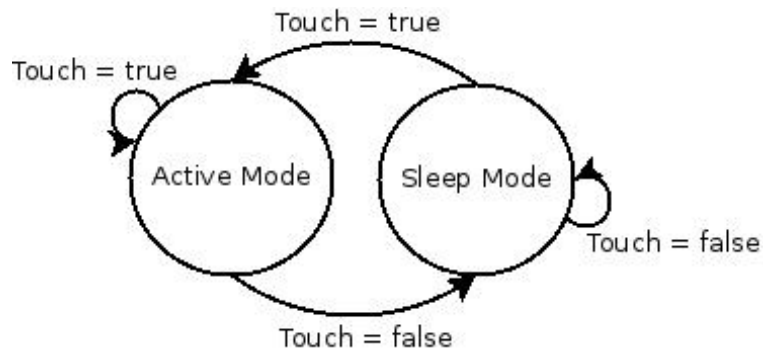


Figure 5.16 The state machine shows how the different modes, the sleep mode and the active mode, are set by the input, touch.

5.3.1 Sleep mode

In this mode, the interface is standing still without any interaction, and this is a very important mode, because it is in this mode it has to attract the user to curiosity and lead them to investigate. In this mode the fan is off. To attract the user to start interacting with the design, the *sleep light* was implemented. The *sleep light* is on medium light intensity and shifts color slowly in a spectrum of light from red to white, and is activated when the design is at rest, e.g. when there is no interaction with the user. Figure 5.17 and equation 5.4 shows how the color changes over time. Having only half of the intensity, it will be even more evident when the modes are changing. By slowly shifting colors it gives the impression of Glownado being in a resting stage, it sleeps. Figure 5.18 shows Glownado in different stages in the sleeping process. The *sleep light* should not shift in a pace that is too fast, since that can negatively affect users that are prone to epilepsy.

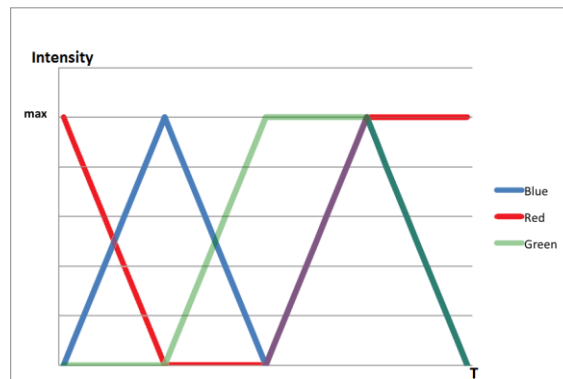


Figure 5.17 The graph illustrates how the intensity of the colors are changing during a time period (T) to give the output color.

The intensity of the colors gives the output color at a certain time, according to;

$$color(t) = red(t) + blue(t) + green(t) \quad [\text{Equation 5.4}]$$

red, blue, green

Intensity represent a number between 0 and max. The max value in the code is 100.

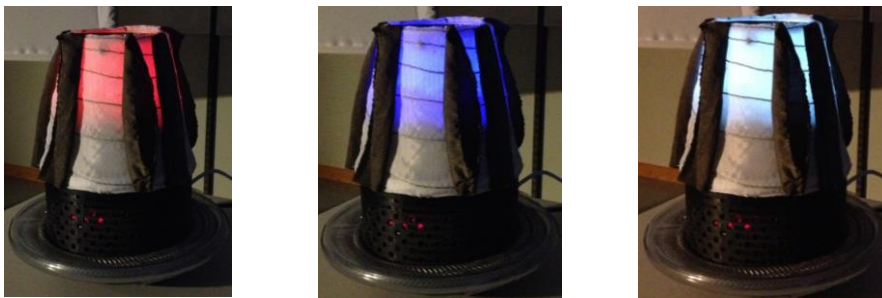


Figure 5.18 In the sleep light the color changes slowly from red to white.

5.3.2 Active mode

The *active mode* is activated when the capacitive sensor detects an approach. In the *active mode* the fan is on, the speed of the fan will increase while the distance to the sensor is decreasing. The *active light* was implemented, to enhance the effect of the fan. The shifting of colors, from the *sleep light*, will stop, and the current color will increase in intensity to its maximum, according to equation 5.5. Too, as mentioned before, clearly notice the change between the modes. When the user no longer interacts with Glownado, the *sleep light* activates and proceeds where it was interrupted.

The color of the *active light* is given by;

$$color(t) = red(t) * weight + blue(t) * weight + green(t) * weight$$

[Equation 5.5]

<i>red, blue, green</i>	Values according to the <i>sleep light</i>
<i>weight</i>	Factor to multiply with to increase the intensity of the colors. The value used in the code was 2.55.

5.3.3 Observations in Snoezelen environment, part 1

The main purpose of the first user interaction was to see how the participants would react to the power of the wind produced by the fan. The mobility of Glownado was going to be observed, to see whether it gives response to gentle touch. The purpose was also to investigate if the *sleep light* was attractive and to get feedback from the staff on the design.

The users interacted with the design in different ways. Selina, figure 5.19, placed the back of the hand or her fingertips against it. By doing this, she could feel the vibration the fan was producing through her body. If the fan was strong she leant her head back to avoid the wind, she also let go of her hand. When the wind had slowed down she immediately started to interact again. According to the staff she enjoyed the strong wind, and she seemed to like Glownado since she was investigating it over and over. Selina could probably not see the light from Glownado. Alfred, figure 5.20, was laying in the waterbed and had Glownado placed either close to his neck or in front of him where he could reach it with his hand. The light was not beneficial for him since he is blind and he could not see it. Due to the waterbed the capacitive sensor was always activated, so it was hard to interact with it in an attended way. Instead, he used his fingers to touch and investigate the fan cover. Ivy, figure 5.21, used her hands to interact and pushed Glownado away when the fan caught her by surprise. At first she did not really understand what had happened, but she got curious and started to investigate more. She laughed a lot every time the wind caught her by surprise. She also seemed to like Glownado, since she started to investigate it more and more, pushing it around.

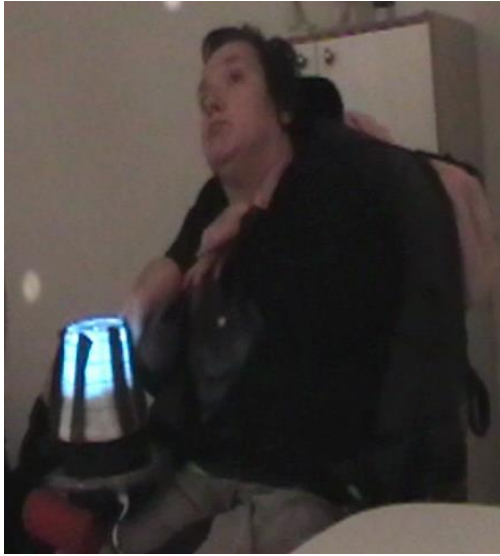


Figure 5.19 Selina used the back of her hand when she interacted with Glownado.



Figure 5.20 Alfred used his fingers to investigate the fan cover.



Figure 5.21 Ivy used her hands to interact with Glownado. When the wind caught her by surprise she pushed it away and laughed.

5.4 Interface Behavior, part 2

After the first observations the reset button, explained in chapter 5.2.5, was added to the control box, since it was noticed as something important and useful.

During the observations it was noticed that having Glownado in a waterbed affected the values of the capacitive sensor. To see if it could be made possible to have the design in a waterbed, tests were made to observe the values of the capacitive sensor when placed in a waterbed. The conductivity of the water was too high, which increased the starting values of the capacitive sensor to a point where it was not possible to compensate in the existing design. This led to the decision that Glownado is not supposed to be placed in waterbeds for now, but can be evaluated for future work.

Many of the users put their hand on top of the fan, but the fan cover was not connected to the capacitive sensor, and did not work as a sensor. A capacitive thread was, therefore, connected to the fan cover. This solution affected the values of the capacitive sensor since the fan cover is made of metal, which made the values unstable.

Another observation was that the light by itself was not enough to trigger the attention of the user, since some users are not able to see the light. A sound, the *sleep sound*, to call for attention was implemented to solve this.

The wind created by the fan was experienced as something fun and exciting for the users during the observations, according to the interpretations of the users' reactions made by the staff. No changes were therefore made to the maximum fan speed.

Two new modes were implemented to the behavior of Glownado, the *wind mode* and the *sound mode*. The state machine shows the new modes, figure 5.22.

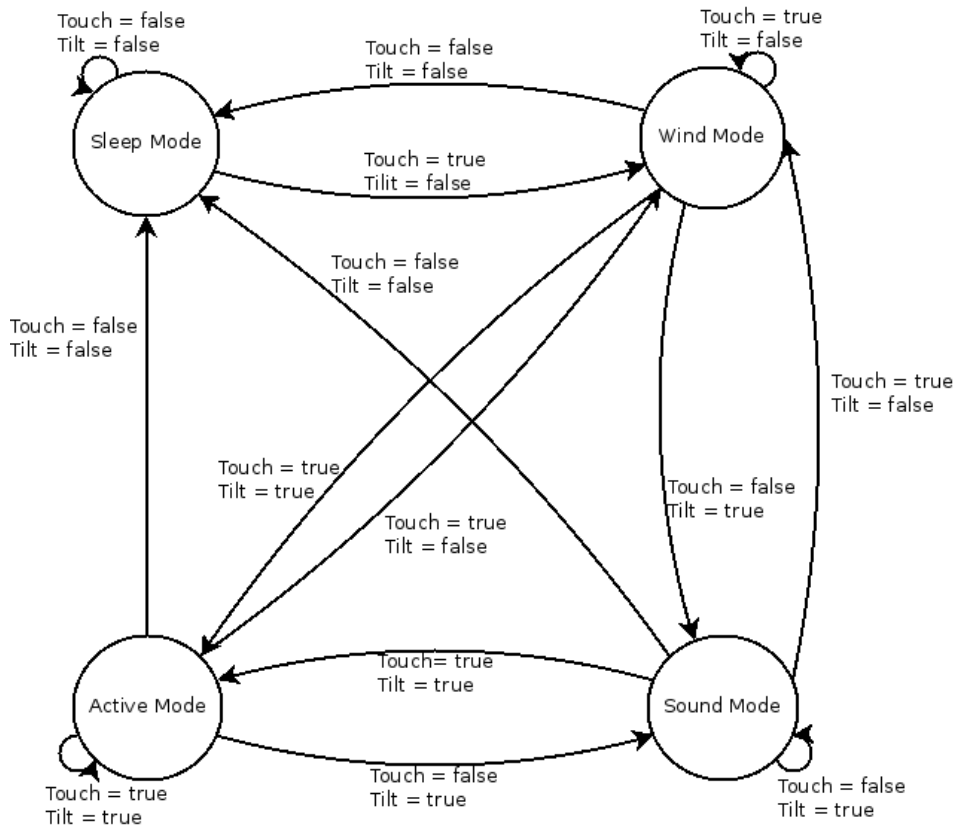


Figure 5.22 The state machine shows how the different modes, the sleep mode, the wind mode, the active mode and the sound mode, are set by the input, touch and tilt.

5.4.1 Sleep mode

The outputs that are being activated in this mode are sound and light. Beside the *sleep light*, explained in chapter 5.3.1, the *sleep sound* will help to call for attention when Glownado is at rest. The notes and the volume are always the same and there is a constant delay between the notes. This was chosen to fit the behavior of the *sleep light*. The sound reminds of something that is breathing in a slow pace.

5.4.2 Active mode

The *active mode* is triggered by touch and tilt, meaning that it activates when the user touches the capacitive sensor and when the user set Glownado in motion. The outputs that are activated in the *active mode* are wind, light and sound. The wind and the light are activated by the capacitive sensor as explained in chapter 5.3.2. The *active sound* was also implemented that activates when Glownado is set in motion, figure 5.23. This is to give the user a response to the motion. The tilting magnitude determines the volume, equation 5.6, and the tilt direction determines the note, equation 5.9. See chapter 5.2.4 for definition of tilting magnitude and tilt direction. This makes it predictable, since the volume always increases when tilted, but unpredictable, since the notes are different depending on how it is positioned, which can arouse curiosity

to explore. The sound type was chosen to give a contrast to the fan, but yet work with the fan so it won't be annoying and create too much impression.



Figure 5.23 In the active mode the wind is blowing, and the sound is activated when Glownado is put into motion.

The volume is given by the tilting magnitude;

$$Volume = \frac{(r - r_{min}) * (Volume_{max} - Volume_{min})}{r_{max} - r_{min}} + Volume_{min} \quad \text{[Equation 5.6]}$$

The note is given by the tilt direction;

$$Note = \frac{(\theta - \theta_{min}) * (Note_{max} - Note_{min})}{\theta_{max} - \theta_{min}} + Note_{min} \quad \text{[Equation 5.7]}$$

r	Tilting magnitude
r _{max}	Maximum tilting magnitude was 20.
r _{min}	Minimum tilting magnitude was 0.
Volume _{max}	Maximum volume used was 127
Volume _{min}	Minimum volume used was 0
Note _{max}	Maximum note was 90.
Note _{min}	Minimum note was 10.
θ	Tilt direction
θ _{max}	Maximum tilt direction was 180.
θ _{min}	Minimum tilt direction was 0.

5.4.3 Wind mode

The *wind mode* is triggered by a touch, meaning that it activates when the user touches or approaches the capacitive sensor. The touch does not set Glownado in motion, meaning that the accelerometer is not active. During this mode, the *active light*, explained in chapter 5.3.2, the *sleep sound*, explained in chapter 5.4.1, and the

wind, explained in chapter 5.3.2 is activated. The *sleep sound* is used in this mode since Glownado is at rest position i.e. not in motion, and the sound should react to the behavior of the motion.

5.4.4 Sound mode

The *sound mode* is triggered by the tilt, meaning that it activates when the user set Glownado in motion. The user does not touch the capacitive sensor or the aura of it, meaning that the capacitive sensor is not active. This occurs when the user touches the lower part of Glownado or the rings around it, or when it continues to swing after being pushed.

The *active sound*, explained in chapter 5.4.2, and the *sleep light*, explained in chapter 5.3.1, are activated during this mode. The *sleep light* is used in this mode since the fan is off and the light reacts to the fan's behavior.

5.4.5 Observations in Snoezelen environment, part 2

As previous user observations, the purpose was to see how the participant would react to the power of the wind produced by the fan, whether the *sleep light* attracts the user, and to get feedback from the staff about the design. The purpose was also to see the reactions to the implemented sound and if it was attractive to the user or not. Once again the mobility of Glownado was going to be observed.

Bane, figure 5.24, used both his hands and arms when interacting with Glownado. It seemed that the light was not bright enough to be noticed by Bane. During this visit, the sound was not acting as intended. The notes were random, and there were no distinction between the *sleep sound* and the *active sound*, i.e. both of the sounds acted at the same time and provided a mix of the two sounds. This occurred depending on the values that was set for the limit that decided when the *sleep mode* and the *active mode* should activate, and also when the vibrations from the fan affected the accelerometer. The staff pointed out that they like the concept of the sound, it was not an irritating sound. It seemed liked Bane was fond of the wind and the vibration from the fan since he kept his hand on Glownado for a long time. His way of showing that he was comfortable with the situation was to vocalize sounds, and he did not bite himself in the arm, which was a sign that he was focused on interacting with Glownado. Bane had problem setting Glownado in motion. The roundness of the bottom was not enough to respond to a gentle push.

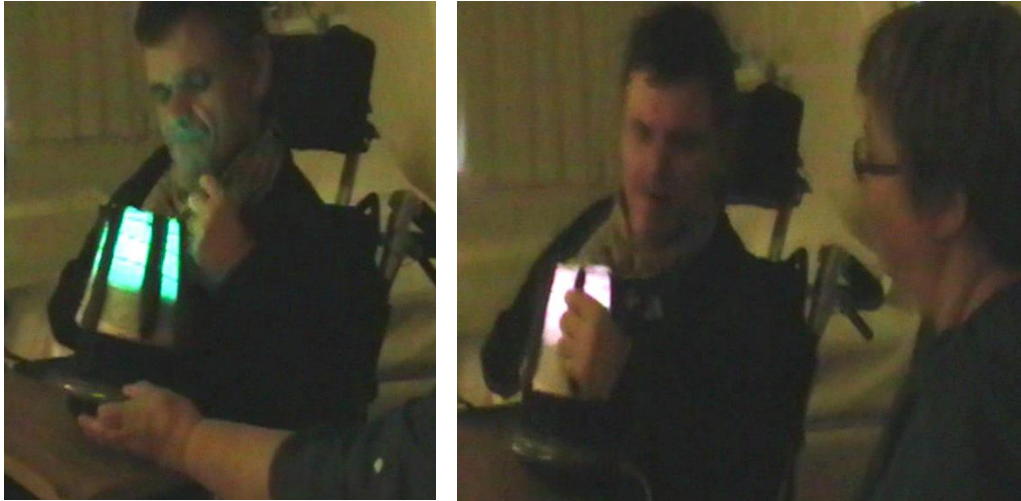


Figure 5.24 Bane used his hands to interact, he held them against Glownado or grabbed the fabric.

5.5 Interface Behavior, part 3

Since the light was not bright enough for Bane to notice, it needed to be intensified. By code, it was not possible to intensify the light, since it was already at its max value. Instead another RGB-LED was added on top of the existing one. A piece of Plexiglas was added between the LEDs to create a distance between them to let more light through the lower one. This increased the light a bit, but not significantly.

During the observation of the interaction between Bane and Glownado it was noticed that the bottom of Glownado was too flat to set in motion. Therefore a layer was added to the bottom to create an even more rounded bottom.

Since the sound was acting randomly during the interaction, there could be no conclusions to whether it was a good way to give a sound response. The sound had to be reconfigured to the next observations.

The state machine was the same as in the previous iteration, figure 5.21, but some attributes in the modes were changed, as explained below.

5.5.1 Sleep mode

The *sleep mode* behaves in the same way as in chapter 5.4.1

5.5.2 Active Mode

The *active mode* follows the same state machine as in chapter 5.4.2, but the code was patched so only the *active sound* would be played in the *active mode* and not randomly the *sleep sound*. An attribute was also added to the *active sound*, if Glownado is tilted in the same position for a longer time the *active sound* will eventually fade out, and change to the *sleep sound*. This was made to draw attention to the user, to make them interact with Glownado more.

5.5.3 Wind mode

The *wind mode* behaves in the same way as in chapter 5.4.3

5.5.4 Sound mode

The *sound mode* follows the same state machine as in chapter 5.4.4, except that the *active sound* was changed as described in chapter 5.5.2.

5.5.5 Other Features

A cinnamon roll was attached to the fan cover at the top of Glownado, to stimulate the sense of smell. When the fan is activated, the wind will spread the scent of the cinnamon roll. Figure 5.25 shows the placement of the cinnamon roll. The cinnamon roll is easy to remove, if the fragrance is not wanted.



Figure 5.25 A cinnamon roll was placed on the fan cover, the fan helps spread the scent.

5.5.6 Observations in Snoezelen environment, part 3

One of the purposes of the visit was to see if the brightness of the light was more noticeable for the users. The observations were also made to see how the user reacted to the sound, whether the sounds would appear interesting in an enticing way that would make the user interact with Glownado. Again the reaction from the intensity of the wind was going to be observed. It was also of interest to see how the users reacted to the cinnamon fragrance, and if the smell was noticeable.

Robin interacted with Glownado for the first time. When he entered the room, he seemed anxious. He interacted with Glownado sitting on the bed in the red room, figure 5.26-28. Immediately he got curious and calm, and started to touch it with his hands. Robin was touching Glownado everywhere, the fan cover, the fabric, the small holes in the bottom and the hoses. He was shaking it, tilted it, and held it tilted in a fixed position. Often he grabbed the conductive fabric to shake it. His look was focused, and it looked like he was thinking of what was happening. If something changed he stopped in that position and reflected about what was happening. During Robin's interaction, he pulled Glownado hard and the power cord suddenly got

unplugged, it was clear that he wondered what had happened and he stopped interacting with Glownado since it stopped being interesting. When the power cord was plugged in again he directly started to interact. After interacting with Glownado for a while, he stopped and gave the impression that he was finished, but as soon as he sat up he started to interact with it again. This time he was less active and he seemed more thoughtful, holding Glownado in the same position for a while.

The sound was turned on and off to see whether he seemed to like the sound or not. Robin did not act any different when the sound was off or when it was on, it was then hard to come to any conclusion about the sound. It was also hard for him to understand that the notes and the volume changed when he moved Glownado, since the changing of the notes were too slow. The cinnamon needed to be damp with water to increase its smell every time after being dried out from the wind of the fan. The fragrance was quite weak, which made it hard for Robin to notice. Instead it provided a tactile effect. The cinnamon roll made him curious and he used his fingers to investigate it. Robin could notice the light from Glownado and he seemed to like the strong wind according to the carer and when he, by himself, could decide to direct the wind towards him.



Figure 5.26 Robin shook Glownado by grabbing the rings.

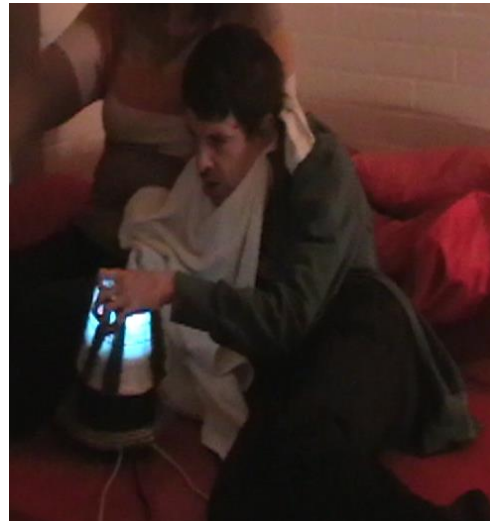


Figure 5.27 Robin used his fingers to investigate the fan cover.



Figure 5.28 Robin grabbed the fabric to set Glownado in motion.

5.6 Interface Behavior, part 4

In previous observation, the sound was hard to control since the fan created vibrations, which affected the values of the accelerometer significantly, this made the changes of the notes random and slow. To mitigate this, the placement of the accelerometer was changed, and was placed under the lid of the Arduino box, instead of the top of the fan. According to the observation the volume of the sound also needed to be intensified, since the noise from the fan overpowered the sound. It also needed to give a quicker response to the users' action, since Robin seemed to not understand the concept of the changes of the notes.

Two different sound types for the *sleep sound*, and two different sound types for the *active sound*, were added which made it possible to appeal to a wider group of users. One of the sound types might be appealing while the other isn't. One sound type was chosen to be arousing and the other more soothing. Sound type 1 is more soothing. It uses a sound that reminds of breathing as the *sleep sound*, and a bass sound as the *active sound*. Sound type 2 is more arousing. It has a high pitch sound as the *sleep sound* and a sound that reminds of church bells as the *active sound*. A switch was added to the control box. It made it possible to choose between different sound types, to turn the volume off, to have both the *sleep sound* and the *active sound* activated, or to only have the *sleep sound* activated, figure 5.29.

The modes were changed to improve the behavior of Glownado to make it predictable but yet unpredictable for the users. The new improvements imply that there will always exist the *active sound* when interacting with the fan, in the same way, as there is the *active light*. This, to give the same response with the sound as with the light, to users with impaired vision. Figure 5.30 shows the new state machine.



Figure 5.29 With the control box the GlowNado can be reset, the sensitivity of the capacitive sensor can be adjusted and the different settings for the sound can be chosen.

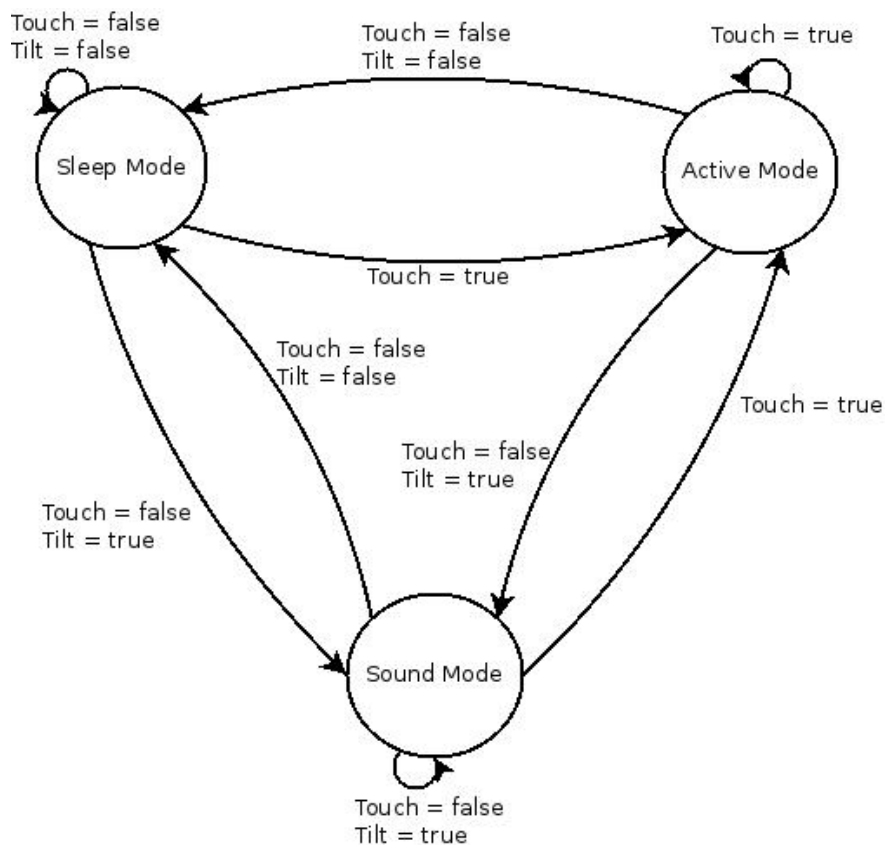


Figure 5.30 The state machine shows how the different modes, the sleep mode, the active mode and the sound mode, are set by the input, touch and tilt. The wind mode was removed.

5.6.1 Sleep mode

The *sleep mode* behaves in the same way as described in chapter 5.4.1.

5.6.2 Active mode

The *active mode* is active when Glownado is sensing a touch, meaning it will always be active as soon as the capacitive sensor detects an approach or a touch. The wind and the light behave in the same way as previous. The *active sound* was changed to better fit the behavior of Glownado. The capacitive sensor controls both, the fan speed and the sound volume. The volume was set according to equation 5.8. This is to make the sound more noticeable, since the noise from the fan increases when the fan speed increases. The tilting magnitude sets the notes, the larger the tilting magnitude, the higher the note, according to equation 5.9. This makes it predictable, as the notes will always increase with the tilting magnitude, yet unpredictable, since it's hard to predict the exact notes, which can arouse the curiosity to explore. When held in the same position the delay between the notes will increase. This is to trigger the user to start, or continue moving Glownado around.

The volume is given by the fan speed;

$$Volume = \frac{(fanSpeed - fanSpeed_{min}) * (Volume_{max} - Volume_{min})}{fanSpeed_{max} - fanSpeed_{min}} + Volume_{min} \quad \text{[Equation 5.8]}$$

The note is given by the tilt magnitude;

$$Note = \frac{(r - r_{min}) * (Note_{max} - Note_{min})}{r_{max} - r_{min}} + Note_{min} \quad \text{[Equation 5.9]}$$

fanSpeed	Fan speed
fanSpeed _{max}	Maximum fan speed was 255.
fanSpeed _{min}	Minimum fan speed was 0.
Volume _{max}	Maximum volume used was 75
Volume _{min}	Minimum volume used was 30
Note _{max}	Maximum note was 70.
Note _{min}	Minimum note was 35.
r	Tilting magnitude
r _{max}	Maximum tilting magnitude was 20.
r _{min}	Minimum tilting magnitude was 0.

5.6.3 Sound mode

The *sound mode* is active when Glownado is sensing a motion, meaning that the accelerometer is active but the capacitive sensor is not. The fan is off in this mode and the *sleep light*, explained in chapter 5.3.1, and the *active sound*, described in chapter 5.6.2, is active.

5.6.4 Other Features

A lightweight fabric was attached to the fan cover. This to provide a visual effect and attract the user to the fan, since wind is not visible the lightweight fabric will reveal the wind and hopefully arouse curiosity. The lightweight fabric is easy to remove, if the feature is not wanted.

5.6.5 Observations in *Snoezelen* environment, part 4

The main purpose of this visit was to observe the user's reaction to the wind, if the light was strong enough to be notice, and to see how the user responded to the sounds. Also to see the effect of the lightweight fabric that was attached to the fan cover. The movability of Glownado was going to be observed as well.

Bane, figure 5.31, held a steady grip of Glownado during the whole interaction. This time he could set Glownado in motion. The capacitive sensor was not behaving as intended, it was random when it was activated, which turned the fan on and off randomly. For this session, it was a good thing, since he held it close to his face the fan became too powerful. Bane could not see the lightweight fabric, but when the wind started to blow the fabric was set in motion and tickled him on the nose and the cheeks. This he really seemed to enjoy. The different sound types and settings were tested. It was hard to tell if he liked the sounds, but according to the staff he probably liked sound type 1 since of experience he hasn't been fond of too low or too high pitch. He has been more comfortable with something in between. A good way to make the sound more noticeable for him was to turn it on and off. Bane then became thoughtful and more aware of the sound. It seemed like he could notice the light from Glownado, since he focused his eyes on it once in a while. It was hard to tell if Bane liked Glownado while he was interacting with it, but he was holding it all the time and didn't want to let it go. Every time the staff was going to take it away he was struggling to keep it. According to the staff this implied that he enjoyed Glownado.



Figure 5.31 Bane held a steady grip of Glownado. The wind made the lightweight fabric spin and tickled his face.

6 Conclusions and Future Work

In general Glownado got much positive feedback from the staff at all three of the Snoezelen institutions. They had common opinion that it was not age dependent, due to the pure design with its simplicity, the natural choice of colors and its shape. This chapter provides conclusions based on the observations and the feedback given by the staff. The focus of the observations was to see how the user reacted to the behavior of the interface. If it was easy for them to interact with it, if they felt comfortable, and if they could understand that their action is the cause of the response. The staff announced that the users need time to get used to new things, which means that the interaction would have been better if they had tried it out a few times more. The chapter also gives suggestions on how the interface can be improved for future work.

6.1 Conclusions

The goal of this master thesis has been to make the interface appealing to a wide range of users. In existing interactive Snoezelen equipment this is often not possible, since they require a bigger effort from the user to interact. In other words, the user needs to be able to move quite a lot to interact with the interface. For example, the LivelyButton in the SID project can be accessible to a wide range of users, since it can be used with many body parts, and the sensitivity can be adjusted, but it is quite big. Without help from the staff or the carer the user is sometimes not able to reach the surface of it, to interact with it.

In this project, it has been important that the interactive interface allows the user to explore their senses, and make them understand that their actions have an impact on their surroundings. Glownado was implemented to give a gradual response, for the users' to understand that they're the ones that affect the environment. During the observations the conclusions have been that it is more important that it gives a quick response to the users' actions, since many of the user has not been able to understand the effect of the gradual response.

To make the users explore their senses there are different outputs, wind, light and sound, and different tactile structure around Glownado, which together creates a multisensory stimulation. According to the feedback given by the staff, this is beneficial since the multisensory stimulation is localized to one area and doesn't create chaos, compared to the outside world where many things can happen and stimulates too many senses at a time.

Glownado opens up many possibilities for the user to explore and interact in different ways, without being limited, since it can be approached, put into motion, and it has different tactile structure that the user can touch.

By making Glownado small and take advantage of the users' abilities to interact with it, it gets accessible to a wider range of users. With a capacitive sensor, that creates an aura around Glownado, the user gets a response from the wind, the light and the sound as soon as they approach it. It does not require much effort from the user to get a big response to their actions. The aura makes Glownado accessible in many different ways, from the sides and the top. To have the ability to change the threshold of the capacitive sensor, it can be adjusted to a wide range of users. The users who participated in this study used their hands to interact with it, but there are many ways to do it, with the feet or the head. During bodystorming these options were tried out. Placing the head against Glownado, figure 6.1, the vibrations and the wind are perceived.



Figure 6.1 The vibrations and the wind are perceived in the face.

Making the conductive fabric as strips was good. They created a bigger aura as they were not completely attached to the fabric. It also invited users with fine motor skills to explore, when fiddling with the fabric, or maybe even pull it.

With the small design it is easy to place Glownado where the user benefit from it the most. It would have been even better if no cables were connected to Glownado, to improve the mobility. With this design, it was not possible, since the fan runs on 1.14A and batteries might drain too fast because of the high power consumption of the fan. Moreover, the capacitive sensor element works better when connected to physical ground.

The shape creates many opportunities of the mobility, it is possible to spin, make it wobble or even have it in a fixed position, by placing it on a ring, which can be seen in figure 6.2. Having all these options, Glownado can be adapted to the user's needs and abilities. According to the staff, the mobility also inspires to team play.

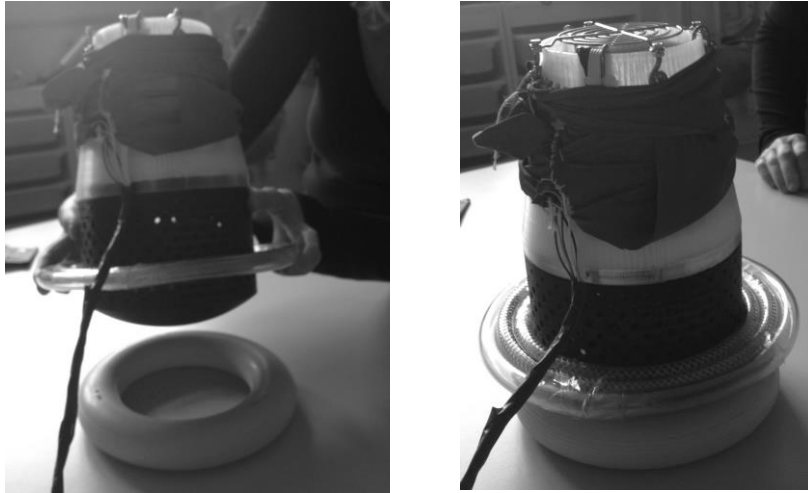


Figure 6.2 By placing Glownado in a ring it stays in a fixed position.

The extra weight, from the metal piece in the bottom of Glownado, was beneficial in more ways than making the design steady, which prevent it from tipping over. It was good, especially for users with impaired vision. It gave a feeling of comfort, when the user could feel and experience it. This comfortable feeling increases the confidence to continue exploring.

The fan cover was used as a protection, but it also gave the user, with fine motor skills, more opportunities to explore Glownado. The cinnamon roll did not provide any noticeable fragrance, when it dried out too fast, to stimulate the sense of smell. Instead it provided a tactile effect, having it on top of the fan were it was possible to touch.

The feedback from the staff was that the light had a good speed when shifting between the colors and it was a good way to create an interest to Glownado. Having shifting colors is a better way to create an interest to start explore, than to only use one color, although the gradual shifting between the colors might not be noticed by some users.

If the user has impaired vision, the option to activate sound was good to lure and tell the user where Glownado is, to enable the user to start interact. By having a sound in the *active mode*, it provides an extra sensory stimulation to the user. This extra sound can be confusing for the user, since both the sound and the noise stimulate the hearing at the same time. During the observations this was never noticed as a problem.

How a person reacts to different sounds is very individual. One person may associate one sound to something negative while, for another, it can be something positive. Having the possibility to choose different sound types or even turning the sound off makes Glownado appeal to a wider range of users.

The option to add lightweight fabrics on top of the fan, to provide a visual effect, might arouse the curiosity of the user and lure them to the wind. During the observations, it was noticed that this feature was also of significance for users with visual impairment. In this case it gave a tactile stimulation and boosted the feeling of the wind, instead of giving a visual effect.

To make the fan interactive contributes to innovation, since this has never been used in interactive Snoezelen equipment before. The wind provides a different tactile

effect, when it is not visible. A sudden gust of wind was surprising and took the user off guard for a second but after realizing what happened it was fun, exciting and adventurous, according to the staff. The surprise element made the users curious to continue exploring. A concern was if the noise of the fan was too loud. According to the staff, this was not a problem. The sound created by the fan can give the user a feeling of security, since the motor sound reminds of something familiar, e.g. a car.

With its charming glow and powerful fan, the words “glow” and “tornado” was combined to form the name of Glownado.

To conclude, Glownado allows the user to experience the concept of cause and effect interactively. By stimulating different senses Glownado can arouse the users' curiosity and allow them to use their creativity to explore in different ways. By giving an immediate response to their actions, it is easier to understand what action caused the outcome. This will give the user the confidence to continue exploring and also give the feeling of having control.

6.2 Future Work

This section describes some considerations to improve the use of Glownado. Solutions to these considerations are proposed for future implementation.

6.2.1 Improvement of the Capacitive Sensor

There were a few problems with the capacitive sensor, which can be investigated in future work. The fan covers and the screws are made of metal. When the capacitive sensor is exposed to metal, the change of capacitance might not be registered. To reduce this problem, the fan covers and the screws can be made of plastic instead. The DC-motor creates fields, e.g. magnetic field, which influences the values of the capacitive sensor. This can be hard to reduce. Different fans can be evaluated to see if there is a difference regarding influence on the capacitive sensor.

6.2.2 Reducing Vibrations

The fan creates vibrations, which affects the values of the accelerometer. A suggestion to how the vibrations can be reduced is to place e.g. some foam rubber between the accelerometer and the surface. Another solution is to use a gyroscope instead of an accelerometer, since a gyroscope is less sensitive to vibrations.

6.2.3 Controlling the Wind

In the last observation, the fan was too powerful for the user. Implementing a control to regulate the fan speed would make Glownado more adaptable when the fan speed can be set to its maximum for users who likes to be exposed to a strong wind, or change the fan speed to be lesser to not make it unpleasant for the user if the wind is too strong.

6.2.4 Implementing a Scent

6 Conclusion and future work

The fan made the cinnamon roll dry out almost immediately, which made the scent difficult to notice. A way to implement a more distinct scent could be to add fragrance to a piece of fabric and attach it to the fan cover.

References

Abras, C., Maloney-Krichmar, D., and Preece, J. (2004). *Bainbridge, W. Encyclopedia of Human-Computer Interaction*. "User-centered design." Sage Publications, Thousand Oaks. 37.4: 445-56.

American Association on Intellectual and Developmental Disability, aaid (2013). *FAQs on Intellectual Disability* (Electronic)
<http://aaid.org/intellectual-disability/definition/faqs-on-intellectual-disability>
[4 Feb 2014]

Arduino (2014). (Electronic)
<http://arduino.com> [16 Jan 2014]

Banner engineering (2014). *Ultrasonic basics* (Electronic)
<http://www.bannerengineering.com> [21 Feb 2014]

Banzy, M. (2009). *Getting Started with Arduino*. Make:Books, an imprint of Maker Media a division of O'Reilly Media Inc., Sebastopol, USA

Buch, B. (2006). *Salutogenesis and Shamanism*. Master Thesis, Department of Health Psychology and Health Education, Göttingen

Centers for Disease Control and prevention (2013). *Facts about Developmental disabilities* (Electronic)
<http://www.cdc.gov/ncbddd/developmentaldisabilities/facts.html> [13 Jan 2014]

Creative Tools (2014). *About 3D-printing* (Electronic)
<https://www.creativetools.se/3d-skrivare-info> [20 Feb 2014]

Elektro:kit (2014). *Fläkt* (Electronic)
<http://www.elektrokit.com> [21 Feb 2014]

Fritzing (2014). (Electronic)
<http://fritzing.org> [25 March 2014]

Gentofte Kommune (2014). *Snoezelhuset* (Electronic)
<http://snoezelhus.gentofte.dk> [20 Feb 2014]

How stuff works (2014). *Kinect sensor* (Electronic)
<http://electronics.howstuffworks.com/microsoft-kinect2.htm> [9 Feb 2014]

Johansson, Lone, occupational therapist at Safirens Upplevelsehus, Malmö, personal conversation 28 August 2013

References

- Kaplan', S and Garrick, B.J. (1981). *On The Quantitative Definition of Risk*, Risk Analysis, Vol. 1, No.1
- Mertens, K. (2008). *Snoezelen - in action*. Shaker Verlag, Aachen, Germany
- Martin, G. (2011). *Introduction to Sound Recording*. (Electronic)
http://www.tonmeister.ca/main/textbook/intro_to_sound_recordingch3.html
[14 March 2014]
- National Center on Accessible Instructional Materials (2013). *Cognitive disability* (Electronic)
<http://aim.cast.org/learn/disabilityspecific/cognitive> [4 Feb 2014]
- Nielsen, J. (1993). *Usability Engineering*. Morgan Kaufmann Publishers Inc., San Francisco, USA
- Noctua (2014). *Fan NF-P12 PWM* (Electronic)
<http://www.noctua.at> [21 Jan 2014]
- Null, L. & Lobur, J. (2006). *The essentials of Computer Organization and Architecture*. 2nd edition. Jones and Barlett, Sudbury, USA
- Pagliano, P. (2012). *The Multisensory Handbook: A guide for children and adults with sensory learning disabilities*. Routledge, New York, USA
- Pink or blue (2014). (Electronic)
<http://www.pinkorblue.se> [14 March 2014]
- Remdahl-Andrén, I. & Malmgren, M. (2008). *Utformning av vårdmiljön - Färgen och estetikens betydelse för välbefinnandet*. Bachelor thesis. Institute of Health and Care Sciences. Gothenburg University, Gothenburg.
- Rompa. (2012). (Electronic)
<http://www.rompa.eu> [13 March 2014]
- Schleicher, D. Jones, PH. & Kachur, O. (2010). *Bodystorming as Embodied Designing, Vol.17, No.6*
- Scythe (2014). *GentleTyphoon* (Electronic)
<http://www.scythe-eu.com> [16 March 2014]
- Sharp, H., Rogers, Y. & Preece, J. (2007). *Interaction Design - beyond human-computer interaction*. 2nd edition. John Wiley & Son Ltd, West Sussex, England
- SID - Sinnlighet, Interaktion, Delaktighet. (2014). (Electronic)
<http://sid.desiign.org> [19 Feb 2014]
- Smith, B. K. (2014). *Bodystorming Mobile Learning Experiences*, Vol. 58, No. 1.
- Sunon (2011). (Electronic)
[http://www.tme.eu/en/Document/18512b03ca83de0f8939fbb59e1e6645/\(D09022860G-00\)-1.pdf](http://www.tme.eu/en/Document/18512b03ca83de0f8939fbb59e1e6645/(D09022860G-00)-1.pdf) [21 Jan 2014]
- Sparkfun (2014). (Electronic)
<https://www.sparkfun.com> [10 Feb 2014]

Spire (2014). *System-cooling* (Electronic)
<http://www.spire-corp.com/system-cooling> [21 Jan 2014]

Wikipedia (2013). *Polära koordinater*. (Electronic)
<http://sv.wikipedia.org> [14 March 2014]

Yourdon, E. & Constantine, L.L. (1979). *Structured Design - Fundamentals of a Discipline of Computer Program and System Design*. Prentice-Hall, INC. Englewood Cliffs, New Jersey, USA

Appendix A: Materials

This appendix describes the different materials used in the project. It presents the different hardware and software, the inputs and outputs that control and give a response to the interactive interface. It also provides information about the physical material used.

A.1 Controller

The controller is an open source platform. With different sensors connected to the input pins, the Arduino can control different actuators on the output pins. It has a 5 V pin, a 3.3 V pin, a VIN (voltage input) pin, and a reset pin. The Arduino board operates at 5 V and maximum 40 mA. It can be powered by a battery or a USB-cable with an input voltage of 7-12 V which is regulated on the board to provide 5 V to the pins, or 3.3 V to the 3.3 V pin. The VIN pin is used if an external power supply is being used. The reset pin can be used to reset the microcontroller. The Arduino board also has internal pull-up resistors of 50 k Ω . For more details about the Arduino board see datasheet (Arduino, 2014).

The libraries in Arduino IDE are written in C/C++, but the programmer only have to define two functions, the setup() and the loop(), to make the programs run. In the setup function, the initial environment is set, and this function only runs once when it is initialized. The loop function, on the other hand, runs after the setup and continues until the board is powered off. When the code is uploaded to the board, it is translated to C/C++ and then compiled into machine-readable instructions that the microcontroller can understand (Banzi, 2009, p.22).

A.2 Capacitive sensor

There are many ways to construct a capacitive sensor. This section describes one type of capacitive sensors, capacitive continuous sensor.

The capacitive continuous sensor uses an RC circuit, which is constructed by connecting a high resistor between two of the pins on the Arduino board - a sensor pin and a receiver pin, and a capacitive material e.g. foil as the capacitor. When the human body cuts the electrical field, it changes. This capacitive sensor measures the time it takes for the voltage of the receiver pin to change to the same state (voltage) as the sensor pin. When the receiver pin has reached its value, the sensor pin changes states. Figure A.1 shows how the sensor pin state changes due to the receiver pin. To make the values more stable a capacitor is connected in parallel with the human body. By connecting a capacitor in parallel, the capacitive value decreases, which lower the

range and thereby makes the circuit more stable. It is also important to connect the Arduino board to ground, since the human body is connected to ground. They have to have the same ground potential, to make the sensor work properly. To make the circuit even more stable, a small floating capacitor can be connected to the receiver pin.

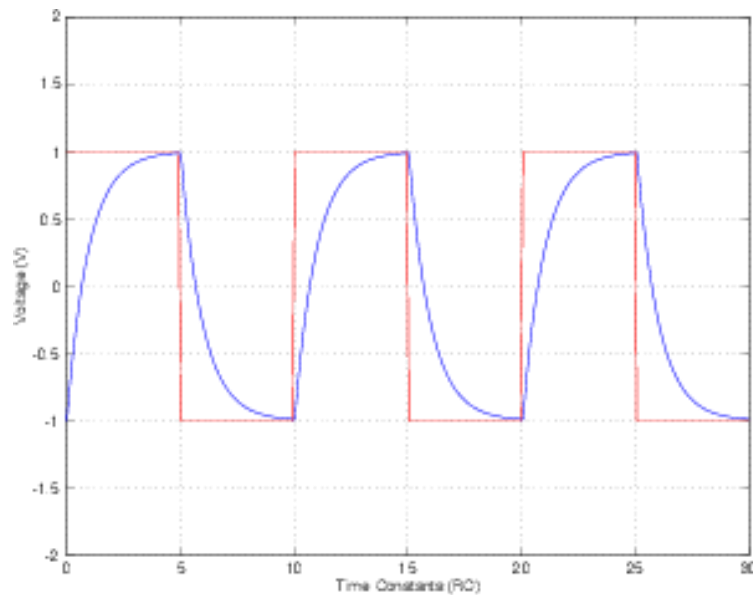


Figure A.1. The y-axis shows the voltage and the x-axis the time. The red line illustrates the state of the sensor pin and the blue line the value of the receiver pin. When the receiver pin reaches a threshold close to the value of the sensor pin, the sensor pin changes state (voltage) (Martin, 2011).

Changing the resistor changes the sensitivity of the sensor. A larger resistor will give a more sensitive sensor since it can detect a change in electricity at longer distance. The drawback of having a large resistor is that the time for a state to change will be longer, which leads to a slower process. If the setup is exposed to metal the sensor pin might not detect the change from the receiver pin (Arduino, 2014).

A.3 Accelerometer

The accelerometer measures both dynamic and static acceleration. Dynamic acceleration is measured when the accelerometer is being shocked or vibrated. The static acceleration senses orientation in relation to the gravity. For more information see data sheet (Sparkfun, 2009).

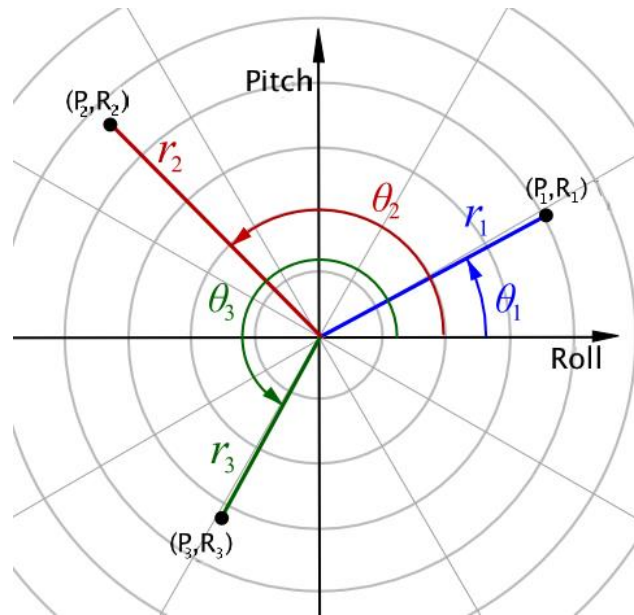


Figure A.2 The figure shows the tilting magnitude (r) and tilt direction (θ) in a two-dimensional coordinate system. P is the pitch angle and R is the roll angle. (Wikipedia, 2013).

A.4 Other Sensors

Another sensors were tested, such as pressure sensor, ultrasonic sensor, capacitive binary sensor and Kinect.

Pressure sensor is a resistor, which changes its resistance when being pressed. The resistance will decrease according to the amount of the pressure. This sensor can be used to measure the force of the pressure.

An ultrasonic sensor measures distance by emitting sound waves. The sound waves will bounce on the specific target and the echo returns to the sensor, to process the sound wave by converting it back to electrical energy. The time measured, when the echo is going back and forth, is proportional to the distance (Banner engineering, 2006).

The capacitive binary sensor, MPR121 is a breakout board that can sense capacitive touch and is driven by an I²C interface. For communication, it only needs to connect two lines to the Arduino. The breakout board has twelve electrodes, to detect touch. It works similar to the capacitive continuous sensor. It discharges and recharges a built-in capacitor and measures the voltage. It detects touch by comparing this voltage with the output voltage of each electrode. If the voltages are not the same a touch is detected, for the specific electrode. See datasheet for more information (Sparkfun, 2010).

Kinect is a game console peripheral that can be used to measure positions and movements. The Kinect uses a Color VGA video camera, a depth sensor and a multi-array microphone. With the depth sensor, it can measure the room in 3D, and with the

array of four microphones it can separate the user from noise. The user can be a couple of meters away from the sensor, and the Kinect can still detect the voice. The sensor can detect and track 48 points of the users' body to model a skeleton structure, the body shape and even the details of the face (How stuff works, 2014).

A.5 Light

The IKEA RGB-LEDs contains three different colors, red, blue and green. The IKEA RGB-LED needs to be supplied by 12 V. The Arduino board can only supply 5V. A driver can be used to supply the RGB-LED with 12 V, without affecting the Arduino board. The signal on the Arduino board, to control the LED, is connected to the input on the driver and the output on the driver is connected to the RGB-LED, now supplied with 12 V.

A.6 Sound

The active speaker used in this project is Roxcore portable speaker. It is quite small, 80x80x30 mm, and 80x80x42 mm when extended. Expanding the speaker improves the quality of sound. It has a built-in rechargeable battery that last for 6 hours.

A.7 CAD/3D Printing

The printer that was used in the project is using a fused filament fabrication (FFF), which is a technique where molten plastic is added layer by layer until the whole object is created. The plastic is added in a way that it creates a grid pattern. This makes the printed object durable (Creative Tools, 2014). The software application used was AutoCAD2013, which is developed and marketed by Autodesk, Inc.

A.8 Conductive Fabric

A conductive fabric, or conductive thread, is a fabric that can conduct electricity. When the capacitive fabric is connected to a source the conductors in the fabric will be filled with charges until the chargers will stay in equilibrium that is when the plates have the same potential as the source. While the conductor is being filled with chargers, it'll create an electric field. When an external object, let's say a finger, cuts into the electrical field, the charges changes.

A.9 Other Materials

Other materials that were used to make the interactive interface are:

- White silk fabric
- Two fan covers, 120x120 mm and 80x80 mm
- A piece of metal, diameter 100 mm, height 15 mm
- Three acrylic glass, 120x120 mm and 50x50 mm
- Velcro tape
- Water hose
- Lamp shade holder
- Arduino box

- Screws and screw-nuts
- Cinnamon roll
- Lightweight fabric

Appendix B: Tables

B.1 Fans

The table shows the different fans tested in the project.

Fan	Dimension (mm)	PWM	Current (A)	Air flow (CFM)	Noise level (dBA)
Sunon EE92251S1-000U-A99 (Sunon, 2011)	120x120x25	No	0.165	51.5	34
Noctua NF-P12 (Noctua, 2014)	120x120x25	Yes	0.05	54.33	19.8
Hair dryer MarQuant 1200W	60x60x60	No	-	-	-
Spire LED SP12025S1H4-B-LED-2P (Spire, 2014)	120x120x30	Yes	0.2	67	21
Everflow B127515BU (Elektro:kit, 2014)	75x75x15	No	0.8	11.83	51
GentleTyphoon D1225C12BBAP-31 (Scythe, 2014)	120x120x25	Possible to add	1.14	150.1	50.5

B.2 Risk Analysis

The risk analysis table shows the different risks, and the calculated risk factor before each observation.

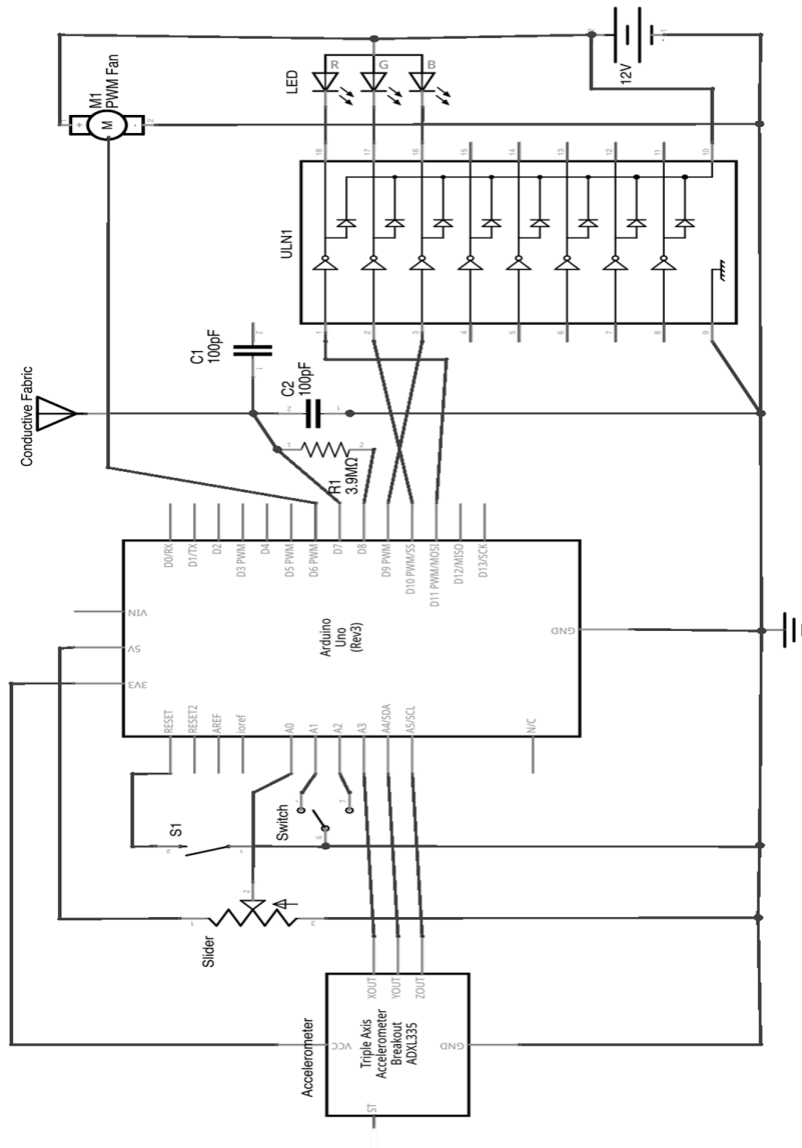
Appendix B: Tables

	Observation, part 1			Observation, part 2			Observation, part 3			Observation, part 4		
	Probability	Consequence	Risk factor	Probability	Consequence	Risk factor	Probability	Consequence	Risk factor	Probability	Consequence	Risk factor
Possible outcome Screws can't keep the top and the bottom together	1	8	8	1	8	8	1	8	8	1	8	8
Fabric gets loose	1	3	3	1	3	3	1	3	3	1	3	3
Capacitive sensor don't work as it should	3	6	18	4	6	24	4	6	24	4	6	24
Jumper wires from Arduino gets loose	3	10	30	2	10	20	1	10	10	1	10	10
The power cord gets loose	2	3	6	2	3	6	2	3	6	1	3	3
Fan cover gets loose (top)	1	10	10	1	10	10	1	10	10	1	10	10
Fan cover gets loose (bottom)	1	10	10	1	10	10	1	10	10	1	10	10
Jumper wires get trapped in the fan	2	10	20	1	10	10	1	10	10	1	10	10
Things fall into the fan	1	10	10	1	10	10	1	10	10	1	10	10
Fan gets loose from the attachment	1	10	10	1	10	10	1	10	10	1	10	10
Fan stops working	1	7	7	1	7	7	1	7	7	1	7	7
LED stops working	1	7	7	1	7	7	1	7	7	1	7	7
Speaker stops working	-	-	-	3	7	21	3	7	21	3	7	21
Speaker gets loose	-	-	-	2	1	2	4	1	4	4	1	4
Sharp edges exposed on Glowradio	1	9	9	1	9	9	1	9	9	1	9	9
Pieces in the bottom gets loose	1	3	3	1	3	3	1	3	3	1	3	3
Glowradio breaks (the plastic)	1	10	10	1	10	10	1	10	10	1	10	10
The hoses around Glowradio fall off	3	3	9	3	3	9	2	3	6	2	3	6
Lightweight fabric falls off	-	-	-	-	-	-	-	-	-	1	1	1
Arduino box gets loose	1	6	6	1	6	6	1	5	5	1	5	5
LED becomes loose	3	9	27	2	9	18	1	9	9	1	9	9
The lid on the Arduino box gets loose	1	5	5	2	5	10	3	5	15	2	5	10
The attached cinnamon roll falls off	-	-	-	-	-	-	2	1	2	-	-	-
Cable to the speaker gets loose	1	3	3	1	3	3	3	3	9	3	3	9

Appendix C: Hardware and Software

C.1 Schematic

The schematics of Glownado. Developed using fritzing (Fritzing, 2014).



fritzing

C.2 Code

The code for Glownado. Divided into subcategories, starting with the master code.

C.2.1 Master

/*

Charlotta Viklund Jan 2014

Glownado, Version 9

Glownado components:

- * Music Instrument Shield
- * ADXL335 - accelerometer
- * Accelerometer library
- * Capacitive Sensor library
- * Driver
- * RGB-LEDs
- * PWM controlled fan
- * Switches 2
- * Push button
- * Potentiometer, Slider
- * Resistor 3.9 MOhm
- * Capacitors 2, 100 pF

Usage

*Sleeping mode for light. Input fixed values. Output RBG-Led

*Active mode for light. Input capacitive sensor and current sleep mode color.

Output RGB-Led

*Gradual change of the fans intensity. Output fan speed, input capacitive sensor.

*Sleep mode for sound. Plays a not with set delay. Input fixed values.

Output Speaker.

*Active mode for sound. Changing notes according to different position.

And volume according to fanSpeed.

Input capacitive sensor value (Fan speed) and accelerometer. Output Speaker.

Based on:

CTaylor (Dec 2011), Musical instrument shield sample sketch, Sparkfun Electronics [Source code],
<https://www.sparkfun.com/tutorials/302>

Paul Badger (2008), CapacitiveSense Library Demo Sketch (V.04), Arduino Playground [Source code],
<http://playground.arduino.cc/Main/CapacitiveSensor?from=Main.CapSense>

Hector Caltenco (2012), ADXL33X Accelerometer Library (commit 41120454b8), SID [Source code],
<https://github.com/hcaltenco/SID-Source/tree/master/libraries/arduino/ADXL330>
*/

```
#include <CapacitiveSensor.h>
#include <SoftwareSerial.h>
#include<Accelerometer.h>
```

```
SoftwareSerial mySerial(2, 3); // RX, TX. but we don't use RX
byte resetMIDI = 4; //Tied to VS1053 Reset line
// Pin definitions
const int redLed = 11;
const int blueLed= 10;
const int greenLed = 9;

const int fanPin = 6;
const int potPin = A0;

const int switchPin1 = A2;
const int switchPin2 = A1;
const int switchPin3 = 12;
```

Appendix C: Hardware and Software

```
Accelerometer accel = Accelerometer(3, 4, 5); // x, y, z pin on accel, analog pin
CapacitiveSensor cs_8_7 = CapacitiveSensor(8,7); // 7 is sensor pin. Resistor between
pin 7 and 8.
```

```
boolean touchSensor=false;
```

```
boolean tiltSensor=false;
```

```
boolean start = true;
```

```
void setup(){
```

```
  Serial.begin(9600);
```

```
  // Setup switches
```

```
  pinMode(switchPin1, INPUT_PULLUP);
```

```
  pinMode(switchPin2, INPUT_PULLUP);
```

```
  pinMode(switchPin3, INPUT_PULLUP);
```

```
  // Setup capacitive sensor
```

```
  cs_8_7.set_CS_Autocal_Millis(0xFFFFFFFF);
```

```
  cs_8_7.set_CS_Timeout_Millis(40);
```

```
  // Setup fan
```

```
  pinMode(fanPin, OUTPUT);
```

```
  //Setup Led
```

```
  pinMode (redLed, OUTPUT);
```

```
  pinMode (greenLed, OUTPUT);
```

```
  pinMode (blueLed, OUTPUT);
```

```
  digitalWrite(blueLed,LOW);
```

```
  digitalWrite(greenLed,LOW);
```

```
  digitalWrite(redLed,LOW);
```

```
  //Setup sound
```

```
//Setup soft serial for MIDI control
mySerial.begin(31250);
//Reset the VS1053
pinMode(resetMIDI, OUTPUT);
digitalWrite(resetMIDI, LOW);
delay(100);
digitalWrite(resetMIDI, HIGH);
delay(100);
talkMIDI(0xB0, 0x07, 120); //0xB0 is channel message, set channel volume to near
max (127)
}

void loop(){

/* To calibrate the z-axis when the fan is off.
   Calibration not declared in setup(), since the fan won't slow down enough.
   Instead a delay of 5000 ms, before calibration.
   This loop only runs once, when initiated.
*/
if(start){
  fanOff(); // safety code to make sure the fan is off.
  delay(5000); //long delay to make sure the fan if off
  accel.calibrate(2,67); // calibrate from the z-axis
  start = false;
}

senseMode(); // Check which sensors that are active.

// fan on and light stays on max, sound is active
if(touchSensor ){
  Serial.println("");
  Serial.println("Active Mode");
```

```
    Serial.println("");
    activeMode();
}
// fan off and light is changing, sound is active
else if(tiltSensor){
    Serial.println("");
    Serial.println("Sound Mode");
    Serial.println("");
    soundMode();
}

// fan off, light changes and sound is sleeping
else{
    Serial.println("");
    Serial.println("Sleep Mode");
    Serial.println("");
    sleepMode();
}
delay(1);
}
```

C.2.2 CapSenseValue

```
int potRead;
int threshold;
const int bufferSize=5;

long capSenseValue(){
    static long bufferCapSense[bufferSize]; // avarage of the last 5 measurements.
    return smooth(bufferCapSense, cs_8_7.capacitiveSensor(10), bufferSize);
}

/* running window smoothing function */
```

```

long smooth(long *bufferVektor, long newVal, int bufferSize){
    long bufferVal = 0;
    for (int i = 0; i < bufferSize-1; i++){
        bufferVektor[i] = bufferVektor[i+1];
    }
    bufferVektor[bufferSize-1] = newVal; // sampling 10 times
    for (int i = 0; i<bufferSize; i++){
        bufferVal += bufferVektor[i];
    }
    return bufferVal/bufferSize; // average value
}

```

/ filterValue, low pass filter or avarage filter.*

*Input: currentValue (the current sensed value), preValue (the previous sensed value),
weight (the weight on the current value)*

Output: The filtered value

**/*

```

double filter( double currentValue, double preValue, double weight){
    return (currentValue*weight+preValue*(1-weight));
}

```

```

int setThreshold(){
    const int MINTHR = 50, MAXTHR = 700; // for RESISTOR 3.9 Mohm
    potRead = analogRead(potPin);
    threshold = map(potRead,0,1023,MINTHR, MAXTHR);
    threshold = constrain(threshold, MINTHR, MAXTHR);
    return threshold;
}

```

C.2.3 Wind

```

const long maxSensValue = 1500;
const byte minFanSpeed = 60;

```

```
const byte maxFanSpeed = 255;
long currentSensValue;
long preSensValue;
int fanSpeed = 0;

// set the speed of the fan
void fanOn(){
    //increasing speed of the fan when increasing the currentSenseValue.
    fanSpeed = map(currentSensValue, setThreshold(),maxSensValue, minFanSpeed,
maxFanSpeed);
    fanSpeed = constrain(fanSpeed,minFanSpeed,maxFanSpeed);
    analogWrite(fanPin,fanSpeed);
}

// set the speed of the fan to 0
void fanOff(){
    fanSpeed = 0;
    analogWrite(fanPin,fanSpeed);
}
```

C.2.4 Light

```
double eventCounter=0;
const double nbrOfEvents=100;
const double increase = 2.55;
const boolean DEBUG_COUNT = false;

/*
Changes the color of the LED gradually
*/
void sleepLight(){
    // Reset eventcounter if necessary
    if(eventCounter>4*nbrOfEvents){
```



```
eventCounter=0;
}

//Compare eventCounter to limits and run appropriate function
if(eventCounter<=nbrOfEvents){
    //increase the blue color and decrease the red color
    increaseAndDecrease(blueLed,redLed,eventCounter,  nbrOfEvents-eventCounter,
1);
    if (DEBUG_COUNT)
        Serial.println("0 Count 100");
}
else if(eventCounter>nbrOfEvents && eventCounter<=2*nbrOfEvents){
    //increase the green color and decrease the blue color
    increaseAndDecrease(greenLed,blueLed,eventCounter-nbrOfEvents,nbrOfEvents-
(eventCounter-nbrOfEvents), 1);
    if (DEBUG_COUNT)
        Serial.println("101 Count 200");
}
else if(eventCounter>2*nbrOfEvents && eventCounter<=3*nbrOfEvents){
    //keep green on max and increase blue and red
    maxAndIncreaseDouble(greenLed,redLed,blueLed,eventCounter-2*nbrOfEvents,
1);
    if (DEBUG_COUNT)
        Serial.println("201 Count 300");
}
else if(eventCounter>3*nbrOfEvents && eventCounter<=4*nbrOfEvents){
    //keep red on max and decrease blue and green
    maxAndDecreaseDouble(redLed,greenLed,blueLed,nbrOfEvents-(eventCounter-
3*nbrOfEvents), 1);
    if (DEBUG_COUNT)
        Serial.println("301 Count 400");
}
eventCounter++;
```

```
}
/*
Increase the light of the current color, depending on the value of nbrOfEvents and
eventCounter
*/
void light(){
    //Compare eventCounter to limits and run appropriate function
    if(eventCounter<=nbrOfEvents){
        //increase the blue color and decrease the red color
        increaseAndDecrease(blueLed,redLed,eventCounter,nbrOfEvents-eventCounter,
increase);
        if (DEBUG_COUNT)
            Serial.println("0 Count 100");
    }
    else if(eventCounter>nbrOfEvents && eventCounter<=2*nbrOfEvents){
        //increase the green color and decrease the blue color
        increaseAndDecrease(greenLed,blueLed,eventCounter-nbrOfEvents,nbrOfEvents-
(eventCounter-nbrOfEvents), increase);
        if (DEBUG_COUNT)
            Serial.println("101 Count 200");
    }
    else if(eventCounter>2*nbrOfEvents && eventCounter<=3*nbrOfEvents){
        //keep green on max and increase blue and red
        maxAndIncreaseDouble(greenLed,redLed,blueLed,eventCounter-2*nbrOfEvents,
increase);
        if (DEBUG_COUNT)
            Serial.println("201 Count 300");
    }
    else if(eventCounter>3*nbrOfEvents && eventCounter<=4*nbrOfEvents){
        //keep red on max and decrease blue and green
        maxAndDecreaseDouble(redLed,greenLed,blueLed,nbrOfEvents-(eventCounter-
3*nbrOfEvents), increase);
        if (DEBUG_COUNT)
```

```
Serial.println("301 Count 400");
}
}

//increase one color and decrease one color
void increaseAndDecrease(int increaseLed, int decreaseLed, int input1, int input2,
double scale){
    byte scaled_input1 = constrain(input1*scale, 0, 255);
    byte scaled_input2 = constrain(input2*scale, 0, 255);
    analogWrite(increaseLed, scaled_input1);//value from 0-100*scale
    analogWrite(decreaseLed, scaled_input2);//value from 100*scale-0
}

//increases two colors. Keep one color on max
void maxAndIncreaseDouble(int decreaseLed, int increaseLed1, int increaseLed2, int
input, double scale){
    byte scaled_input = constrain(input*scale, 0, 255);
    byte scaled_max = constrain(nbrOfEvents*scale, 0, 255);
    analogWrite(increaseLed1,scaled_input);//value from 0-100*scale
    analogWrite(increaseLed2,scaled_input);//value from 0-100*scale
    analogWrite(decreaseLed,scaled_max); // this is for the moment not important but
incase the code changes we keep it
}

// decreases two colors. Keep one color on max
void maxAndDecreaseDouble(int increaseLed, int decreaseLed1, int decreaseLed2,
int input, double scale){
    byte scaled_input = constrain(input*scale, 0, 255);
    byte scaled_max = constrain(nbrOfEvents*scale, 0, 255);
    analogWrite(decreaseLed1,scaled_input);//value from 100*scale-0
    analogWrite(decreaseLed2,scaled_input);//value from 100*scale-0
    analogWrite(increaseLed,scaled_max); // this is for the moment not important but
incase the code changes we keep it
```

```
}
```

C.2.5 Modes

```
// fan off, light is sleeping and sound is sleeping
```

```
void sleepMode(){  
    fanOff();  
    sleepLight();  
    sleepSound();  
}
```

```
// fan off and light is sleeping, sound is active
```

```
void soundMode(){  
    fanOff();  
    sleepLight();  
    activeSound();  
}
```

```
// fan on and light stays on max, sound is active
```

```
void activeMode(){  
    fanOn();  
    light();  
    activeSound();  
}
```

C.2.6 SenseMode

```
const boolean ALLOW_FILTER = true;  
const boolean ALLOW_BLADJUST = true;  
const boolean DEBUG_ACCEL = true;  
const boolean DEBUG_CAPSENSE = true;
```

```
const double fwSens = 0.2;  
const double fwAccel = 0.3;  
const double blSlow = 0.055;
```

```
const double blFast = 0.30;

double currentRoll;
double currentPitch;
double preRoll;
double prePitch;
double adjustedCurrentRoll;
double adjustedCurrentPitch;
double rollBaseline;
double pitchBaseline;
double preCurrentRoll;
double preCurrentPitch;

int distance;
int adjustedDistance;

void senseMode(){
  //CapSense
  //Use the value from the capSensValue() and filter it
  preSensValue = currentSensValue;
  currentSensValue = capSenseValue(); // get the smoothed value of capsense
  if (ALLOW_FILTER)
    currentSensValue=filter(currentSensValue, preSensValue, fwSens); // running
    average filter

  // if there is i timeout
  if (currentSensValue < 20)
    currentSensValue = maxSensValue;

  long currentThres = setThreshold();
  if (DEBUG_CAPSENSE) {
    Serial.print("CapSense: ");
```

```
Serial.print(currentSensValue);
Serial.print("\t Threshold: ");
Serial.print(currentThres);
Serial.print("\t Fan Speed: ");
Serial.print(fanSpeed);
Serial.println();
}

// threshold to activate the capSens
if(currentSensValue > currentThres){
    touchSensor = true; // touch if detected
    Serial.println("Touch true");
}else{
    touchSensor = false; // touch not detected
    Serial.println("Touch false");
}

//Accelerometer. Filter the roll and the pitch to even out "random values"
accel.update();

preRoll = currentRoll;
currentRoll = accel.roll();
prePitch = currentPitch;
currentPitch = accel.pitch();

if (ALLOW_FILTER) {
    currentRoll = filter(currentRoll, preRoll, fwAccel); // low pass filter. Lower weight
to the current values gives more stable system
    currentPitch = filter(currentPitch, prePitch, fwAccel); // low pass filter. Lower
weight to the current values gives more stable system
}
```

```
distance = hypotenuse(currentRoll, currentPitch);

if (ALLOW_BLADJUST) {
// Change pitch and roll according to the baseline, adjusted pitch and adjusted roll
double rw, pw;
if(abs(pitchBaseline) < abs(currentPitch)){ // when tilted
    pw = blSlow;
}else{ // When going back to idle
    pw = blFast;
}
if(abs(rollBaseline) < abs(currentRoll)){// when tilted
    rw = blSlow;
}else{ // When going back to idle
    rw = blFast;
}

pitchBaseline = filter(currentPitch, pitchBaseline, pw);
rollBaseline = filter(currentRoll, rollBaseline, rw);
}
adjustedCurrentPitch = abs(currentPitch-pitchBaseline);
adjustedCurrentRoll = abs(currentRoll-rollBaseline);

// to be used as decetor of activation
adjustedDistance = hypotenuse(adjustedCurrentRoll, adjustedCurrentPitch);

if (DEBUG_ACCEL) {
    Serial.print("Roll: ");
    Serial.print(currentRoll);
    Serial.print("\t Pitch: ");
    Serial.print(currentPitch);
    Serial.print("\t Distance: ");
```

```
Serial.print(distance);
Serial.print("AdjustedRoll: ");
Serial.print(adjustedCurrentRoll);
Serial.print("\t AdjustedPitch: ");
Serial.print(adjustedCurrentPitch);
Serial.print("\t AdjustedDistance: ");
Serial.print(adjustedDistance);
Serial.println();
}

if (adjustedDistance > 7){
    tiltSensor = true; // tilt detected
    Serial.println("Tilt true tilt");
}else{
    tiltSensor = false; // tilt notdetected
    Serial.println("Tilt false");
}
}

int hypotenuse(double p1, double p2){
    return (int) sqrt(p1*p1 + p2*p2);
}

//the angle function, converted from rad to degree
int angle(int p1, int p2){
    return (int)(atan2(p1,p2)*180)/M_PI;
}
```

C.2.7 Sound

```
const bool DEBUG_AUDIO = true;
int switch1, switch2, switch3;
```



```
byte volume;
byte note;
byte preNote = 0;
int preDistance = 0;
int noteTime = 5;
int distDiffThres = 5;

long startTime=0;
long soundOnTime = 0;
long soundOffTime= 0;

const byte sleepVolume[] = {20,40};
const byte sleepNote = 60;
const byte minActiveVolume = 30;
const byte maxActiveVolume = 75;
const byte minActiveNote = 35;
const byte maxActiveNote = 70;

int sleepInstrument[] = {126,9};
int activeInstrument[] = {16,14};
int instrument;

void sleepSound(){

    talkMIDI(0xB0, 0, 0x00); //Default bank GM1
    // turn of sound if it is active in active mode
    if(instrument == activeInstrument[0] || instrument == activeInstrument[1]){
        turnSoundOff();
    }
    getSleepInstrument();
    getSleepVolume();
```

```
    sound(instrument, sleepNote, preNote, volume, 2500, 1800);
    preNote = sleepNote;
}

void activeSound(){
    talkMIDI(0xB0, 0, 0x00); //Default bank GM1

    // turn of sound if it is active in sleep mode
    if(instrument == sleepInstrument[0] || instrument == sleepInstrument[1]){
        turnSoundOff();
    }

    getActiveInstrument();

    // The volume increases when the fan sped increases. If the fan is off and the tilt
    // is active there should still be sound.
    getActiveVolume();

    note = map(distance,0,20, minActiveNote, maxActiveNote);
    note = constrain(note, minActiveNote, maxActiveNote);

    if(abs(preDistance-distance) < distDiffThres){
        noteTime += 10;
    }else{
        noteTime = 5;
    }
    noteTime = constrain(noteTime,5, 700);
    Serial.print("NOTETIME: ");
    Serial.println(noteTime);
    Serial.println(" ");
}
```

```
//play sound
    sound(instrument, note, preNote, volume, noteTime, noteTime);

    preNote = note;
    preDistance = distance;

}

// turn on and off notes.
void sound(int instrument1, int note, int preNote, int volume, int soundTime, int
offTime){
    if (note != preNote){ // turn off the sound if the note not is equal to the previous
note.
        noteOff(0,preNote,volume);
        timerOff();
    }
    talkMIDI(0xC0, instrument, 0); //Set instrument number. 0xC0 is a 1 data byte
command
    if(startTime == 0){
        soundOnTime = soundTime;
        soundOffTime = offTime;
        timerOn(); // turn timer on to start the counter
        soundOnTime += startTime; // the time for the note to be on
        soundOffTime += soundOnTime; // the time the note should be off
        noteOn(0,note,volume); // turn on the sound
    }
    timerOn(); // turn on timer to start a new counter
    if(startTime > soundOnTime){ // turn off the note when it has been on for the given
time
        noteOff(0,note,volume);
    }
}
```

Appendix C: Hardware and Software

```
    if(startTime > soundOffTime){ // turn off the counter when the note has been off for
the given time
        timerOff(); // startTime = 0.
    }
    if (DEBUG_AUDIO) {
        Serial.print("Instrument: ");
        Serial.print(instrument);
        Serial.print("\t Note: ");
        Serial.print(note);
        Serial.print("\t Volume: ");
        Serial.print(volume);
        Serial.print("\t Note duration: ");
        Serial.print(soundOffTime-soundOnTime);
        Serial.println();
    }
}

// turn off previous and current sound
void turnSoundOff(){
    timerOff();
    talkMIDI(0xC0, instrument, 0);
    noteOff(0,preNote, volume);
    noteOff(0,note, volume);
}

// get the instrument for the sleep sound
void getSleepInstrument(){
    readAudioSwitch();
    if(switch3 == LOW){
        instrument = sleepInstrument[0];
    }else{
        instrument = sleepInstrument[1];
    }
}
```

```
}  
}  
  
// get the volume for the sleep sound  
void getSleepVolume(){  
    readAudioSwitch();  
    if(switch1 == HIGH && switch2 == HIGH){  
        volume=0;  
    }else if(instrument == sleepInstrument[0]){  
        volume = sleepVolume[0];  
    }else{  
        volume = sleepVolume[1];  
    }  
}  
  
// get the instrument for the active sound  
void getActiveInstrument(){  
    readAudioSwitch();  
    if(switch3 == LOW){  
        instrument = activeInstrument[0];  
    }else{  
        instrument = activeInstrument[1];  
    }  
}  
  
// get the volume for the active sound  
void getActiveVolume(){  
    readAudioSwitch();  
    if(switch2 == LOW){  
        volume = map(fanSpeed, 0, maxFanSpeed, minActiveVolume,  
maxActiveVolume);  
    }else{
```

Appendix C: Hardware and Software

```
    volume = 0;
  }
}

// read switch state
void readAudioSwitch(){
  switch1 = digitalRead(switchPin1);
  switch2 = digitalRead(switchPin2);
  switch3 = digitalRead(switchPin3);
}

// start a counter
void timerOn(){
  startTime = millis();
}

// turn off a counter
void timerOff(){
  startTime = 0;
}

//Send a MIDI note-on message. Like pressing a piano key
//channel ranges from 0-15
void noteOn(byte channel, byte note, byte attack_velocity) {
  talkMIDI( (0x90 | channel), note, attack_velocity);
}

//Send a MIDI note-off message. Like releasing a piano key
void noteOff(byte channel, byte note, byte release_velocity) {
  talkMIDI( (0x80 | channel), note, release_velocity);
}

//Plays a MIDI note. Doesn't check to see that cmd is greater than 127, or that data
values are less than 127
```

```
void talkMIDI(byte cmd, byte data1, byte data2) {
  mySerial.write(cmd);
  mySerial.write(data1);

  //Some commands only have one data byte. All cmds less than 0xBn have 2 data
  bytes
  //(sort of: http://253.ccarh.org/handout/midiprotocol/)
  if( (cmd & 0xF0) <= 0xB0)
    mySerial.write(data2);
}
```