The Brave Little Troll

a visual rhythm game for the Deaf and hearing-impaired children

Juho Jouhtimäki
Aalto University School of Art and Design
Department of Media
Media Lab Helsinki
Master of Arts in New Media
March 2010

Acknowledgements

I thank Finnish Association of the Deaf/OSATA(KNACK)-PROJECT for organising and supporting the project, and RAY (Finland's Slot Machine Association), for financial support of Osata.

This thesis would not have been possible without the continuous support, help and guidance of Dr Päivi Rainò. I owe my deepest gratitude also to my supervisor, Petri Lankoski for excellent guidance with the written thesis. I am deeply thankful to my opponent in the Master's thesis seminar, Jari Suominen, for excellent comments about the work.

I would also like to thank Andrea Botero, Hanna Hyvättinen, Elina Kivelä-Taskinen, Terttu Martola, Antti Mäki, Jari Nyberg, Russ Palmer, Maria Palavamäki, Juha Sarkkola, Antti Raike and Rasmus Vuori for their insightful comments and ideas concerning the game. I owe special thanks to Irja Seilola and her sons Eelis and Ilmo for participating the game test. I appreciate all the practical help Ilpo Kari and Pekka Salonen provided during the project.

I warmly thank Anna Keune for motivational support and comments during the writing process. I also thank Matti Luhtala, Abhigyan Singh and Diana de Sousa for their comments.

I owe my gratitude to Riia Celen and Suvi Kitunen for inviting me to the project and providing the whole team with their knowledge and insights. I thank the rest of the game team: Teemu Korpilahti, Sven Noben, and especially I thank Margaret Plaisted for proofreading and commenting the text.

I thank everyone who participated the Master's thesis seminar with me and shared their wisdom with me. I thank everyone who participated in our interviews at Haukkaranta school in Jyväskylä on the 11th of June, 2008, and everyone who has commented and tested the game in the various events and online. I would also like to thank all other members of the Deaf community for warmly welcoming me as their guest and co-worker.

Table of Contents

1 Preface....1
1.1 Personal background....1
2 Introduction....3
2.1 A peek into the Deaf culture....3
2.2 Developing sense of rhythm....4
2.3 Background of the game....5
2.3.1 Target group....6
2.3.2 Game concept....6
2.3.3 Game story....7
2.3.4 Visual style....8
2.3.5 Sound vs. no sound....8
2.4 Gameplay in some rhythm games....9
3 Developing The Brave Little Troll....15
3.1 Examining technical possibilities....15
3.2 Graphical user interface....17
3.2.1 Main menu....17
3.2.2 Settings menu....18
3.2.3 Game interface....20
3.2.4 Rhythm bar....22
3.3 Level design....24
3.3.1 General level structure....24
3.3.2 Tree level....26
3.3.3 River level....30
3.3.4 Mountain path level....31
3.3.5 Castle wall level....32
3.3.6 Tutorial level....32
4 Finalising....34
4.1 Optimisation....34
4.2 Polishing....34
4.3 Web version....36
4.3.1 Loading animation / preloader....36
4.3.2 Optimisation....37
5 Conclusions....38
5.1 Learning outcomes....38
5.2 The future of the game....38
5.3 Other reflections and outcomes of the project....38
6 References....41
7 Appendix I: Game testing plan....45
8 Appendix II: The public events where the game was displayed....49

1 Preface

The thesis work consists of the following parts:

- *The Brave Little Troll* computer game
- written thesis

The aim of this work is to reflect and document the multiple phases of the design and development process of *The Brave Little Troll*. This paper is a valuable contribution to the design community as it provides a record of our project challenges and solutions. Thus being a useful resource for other project groups facing similar challenges.

The work aims to describe on general level; the project's goals, audience, methods, challenges, solutions, results and limitations. This paper also addresses the future work and application of the game as well as my role and involvement in the project.

1.1 Personal background

I have a BA in interaction design[1] and I have also studied literary arts, philosophy and the Hungarian language and culture. I do not have formal education in software engineering or computer science, but I have gained programming experience through various projects.

I also do not have an education in pedagogy or music theory, however the game would have benefited from experience in both fields. These facts might explain some of the problems I am going to describe in the later chapters.

I have been interested in and been playing computer games since early childhood. In recent years I have been increasingly interested in the application of games for other purposes than pure entertainment. In 2006 I designed and developed an educational game called *Nugali*[2], as the practical part of my BA thesis. Before that I had been working in SIG-GLUE project[3] which aimed to ”promote more and better use of better learning games”.

Nugali is a turn-based game with elements from life simulation games, role-playing games and resource management games. The game is very different from *The Brave Little Troll* and the

1 From TAMK University of Applied Sciences, School of Art and Media
2 The game is not currently on-line, but the thesis is available at https://oa.doria.fi/handle/10024/5261
3 Special Interest Group for Game-based Learning in Universities and life long learning, http://www.sig-glue.net

experience gained from that project did not help much with the challenges of the new one.

2 Introduction

This chapter provides some background information for the project. I explain why *The Brave Little Troll* was created, outline my role in the project, describe the audience the game was made for, explain various design decisions that were made and tie *The Brave Little Troll* into some context both as a game and a tool.

2.1 A peek into the Deaf culture

Before starting in the project I was not familiar with the Deaf[4] culture. I had seen some documentaries, experienced fictional works, watched signed news, and seen people signing. However, I did not have any first hand experience.

With the guidance of Dr. Päivi Rainò, the coordinator of the OSATA[5] project from the Finnish Association of the Deaf, I was able to gain an insight into the Deaf community. When the gates were opened I met many other friendly and helpful people from the community who enlightened me on various aspects. I learned the very basics of sign language, received a signed name, saw a signed opera and heard about the complex issue of deafness, Deaf culture, the cochlear implant children, and the crisis of the sign language.

Suvi Kitunen provided a lot of useful information during the project as she was working on a Deaf culture specific web site, Knack.fi. Kitunen (2009, pp.4) points out the uncertainty of the future of the Deaf society and culture referring to Ladd (2003) and Sacks (1991). The reason for this is that the number of people signing is rapidly decreasing. This is due to the fact that in recent times an increasing number of deaf children are being fitted with cochlear implants[6]. These children are most often born to hearing parents. The need for sign language no longer exists any more, and thus the children do not become full members of the Deaf community.

One of the team members, Sven Noben, is a member of the Deaf community. He provided us a lot of important information on how the Deaf perceive many aspects of life. Instead of perceiving themselves disabled, like many hearing people perceive them, many of them feel the constraints are laid on them by the dominant, hearing culture. Kitunen (2009, pp.4) referring to other researches such as Ladd, points out the Deaf have suffered of a long oppression by the mainstream culture, and

4 Deaf with a capital ‘D’ is used typically in deaf literature to point the group of people who identify themselves with sign languages and deaf culture. Deaf with a small ‘d’ is rather used to refer to medically deaf people.

5 Http://www.osata.net

6 A cochlear implant is a surgically implanted electronic device which provides a sense of sound to a person who is profoundly deaf or severely hearing-impaired.

only recently their needs and rights have been recognised.

As the rest of the team consisted of hearing persons, we tried to listen very carefully the needs of the Deaf community when designing the game. During the project, there were a lot of meetings with the members of the community, and these meetings provided a lot of useful information for many design decisions we made. Whether we were successful in catering the needs, remains to be seen.

2.2 Developing sense of rhythm

The Brave Little Troll is loosely based on existing theories of rhythm learning, and methods of rhythm teaching for the Deaf and hearing impaired. I have not been involved in any research concerning that. However, to put the game into context, I will briefly introduce some related studies and methods.

A study conducted by researchers from Hungarian Academy of Sciences supports a view that beat perception is innate Winkler, et al., (2009). Sense of rhythm develops further after birth. Hearing children have the advantage of perceiving auditory rhythm, but deaf children need to rely mostly on visual and tactile cues. Their sense of rhythm is often underdeveloped when they start school. Poor sense of rhythm has been linked to dyslexia (Goswami, 2008).

The Brave Little Troll aims to improve children's abilities in identifying and producing rhythmic patterns before they commence school. This should support language perception and the early development stages of reading and comprehension, as it has been proven to work for children with dyslexia Kujala, et al., (2001) and Overy, et al., (2003). A research conducted with deaf children by Meronen (2004) suggests also a link between motor abilities, like serial and repeated finger tapping, and written language readiness.

Often, when I have mentioned working on a rhythm game, people have usually first thought it is some kind of music game. It should be remembered, however, that rhythm is not only related to sound. One can find rhythm everywhere. From the pulsing heart in one's chest to dripping water from an unfixed faucet, from waves in the ocean to the stripes on a zebra crossing. *The American Heritage Dictionary of the English Language* (2003) defines rhythm as: "movement or variation characterised by the regular recurrence or alternation of different quantities or conditions."

Terttu Martola, a Finnish rhythm education instructor, has been teaching rhythm for deaf children for decades. Her methods include using vision and sense of touch to absorb tempo and rhythmic structures. Body exercises are used to internalise them. (Martola, 2005).

Naomi Benari, a British dance instructor for the Deaf and hearing impaired, has been teaching dance to deaf children. According to her “the children gradually develop the ability to cope with quite complex rhythms” if the rhythm work is visual, notated and experienced with the whole body. (Benari, 1995, pp.45).

British music therapist Russ Palmer's approach is “Feeling the Music Philosophy”. This means feeling music through vibrations instead of listening to it through ears. He uses this method with sensory impaired and deaf-blind people. (Palmer, n.d.). The team consulted with him during the whole project.

2.3 Background of the game

My background in learning games was the key reason for my initial involvement. In beginning of 2008 I was approached by two Media Lab students, Suvi Kitunen and Riia Celen who had started working for the Osata project. They invited to join the Osata project and briefed me on the project's general concept and goals. The idea was to create a game that could be used by preschool aged children who were Deaf and hearing-impaired. The game would be used as a training tool for developing a sense of rhythm. Any further specifications were left for the game team to work out.

The initial setup of the team changed slightly before the real work started. At that point the team was the following: Sven Noben as project manager, Margaret Plaisted as illustrator and graphic designer, Suvi Kitunen as script writer and concept designer, and Teemu Korpilahti and I as the programmers. During the project the roles and the staffing changed. Teemu's role changed from programmer to Flash animator and I became the main programmer. The core code was already written by me.

Sven Noben moved to Belgium and left the project in January 2009 after we had finished the first level of the game. The coordination of tasks fell into my hands after that quite naturally, because I was melding the art, animations and the code together in the development environment. There are no audio or sound tracks in the game because the game is about visual rhythm. I will return to this in chapter 2.6.

Most of the overall design decisions were made in the team and there is no one person to point out as the game designer. Also the roles overlapped here and there during the project as the team was quite small. Since early 2009 the core team was Margaret Plaisted, Teemu Korpilahti and myself. In September 2009 Margaret moved to Canada and worked remotely from there. We all worked part-

time on the project dividing time between studies and other projects.

The game was published[7] on Finland's first sign language day, February 12th, 2010.

2.3.1 Target group

The target group of the game was set by the client to 4 to 7 year old Deaf and hearing-impaired children whose sense of rhythm is underdeveloped.

The age group is very diverse. There is a huge gap in development between 4 and 7 year old children. The challenge is to keep the game simple enough for the youngest children, but at the same time keep it challenging and interesting enough for the oldest ones.

Suvi Kitunen and I interviewed[8] some parents of 3 to 8 year old deaf children about the children's computer using abilities and habits. Based on the answers, and combined with findings in research conducted on children's computer use within or near the age group (Buckleitner, 2008; Oblinger & Oblinger, 2005), we made several assumptions:

- the children in the target group have necessary motor skills to use a mouse, a keyboard and the Wii Balance Board
- the game should not be very long as the attention span is shorter among the younger children
- the game should not contain violence, and it should not be too scary
- the user interface of the game should be simple and intuitive
- the game should be as polished as possible to avoid frustration caused by bugs and glitches

2.3.2 Game concept

Our aim was to create a game which required the player to follow and reproduce rhythm. We wanted the game to be both entertaining and valuable. We did not intend to compete with commercial products, but we wanted to create something much more appealing than an average educational game. The game was to use visual rhythm and in a future version body exercises to support rhythm learning.

7 The game is downloadable or playable online for no cost at http://www.knack.fi
8 The interviews took place in Jyväskylä at Haukkaranta school (School for the Deaf).

2.3.3 Game story

We wanted the game to have a storyline which would help the children become interested in the game. The story concept and script was largely written by Suvi Kitunen and Margaret Plaisted. I participated in the brainstorming sessions. The story is communicated through non-verbal animations. We decided not to use any written, spoken or signed language for several reasons. The obvious reason for not using spoken word was the target group. Deaf and hearing-impaired children would miss most of it. Using text was also ruled out as most children in the target group would not learnt to read yet. A video clip of a signing person seemed to break the visual context of the game. An animated character signing the story was ruled out partly due to limited resources. We were also hoping to get the game internationally published, so we wanted the game to be easily adapted to different languages and cultures. Translating the game content could be problematic in some cases regardless of it being readable, audible or signed.

The game starts with an introduction animation. The sun rises in a village populated by green trolls. The main character of the game, a little troll, sleeps in the same room with his mother when a vulture flies into the room and steals a jewel from the mother's necklace. This jewel is an orb heart. It contains the inner rhythm of the troll who wears it.

The little troll wakes up and looks out. He sees the village is in chaos. The orb hearts of every adult have been stolen and they can not perform their daily tasks any more. For example, the fishermen can not row the boats, because rowing requires a certain rhythm.

Troll-children still have their orb hearts. Maybe they have not been stolen, because the children's inner rhythm is not developed enough.

The little troll sees that the vultures are taking their loot to a nearby island. They are heading towards a castle in the mountains. The troll decides to chase after the vultures to rescue the orb hearts. He goes down to the shore and gets a boat.

The player guides the troll through different surroundings. First, the troll must row to the island. There he meets a monkey who grabs a fallen orb heart and climbs to a tree. The player must help the troll climb up the tree. When the troll reaches the top, the monkey gives the orb heart to him. From the tree top there is a view of a river and the castle, and one can see there is a long way to go.

The troll must then cross the river by jumping from rock to rock, ride with a befriended emu on a mountain path avoiding deep gorges and finally climb a castle wall to find out where the orb hearts

have been taken to.

The game ends with a closing animation in which the troll meets a wizard who has difficulty with breathing. There is something wrong with the rhythm. That is why the wizard has stolen the orb hearts. He wanted to find a correct rhythm for breathing. The troll suspects there is some other reason behind the wrong rhythm. He hits the wizard's stomach with his head and a frog jumps out of wizard's throat. The frog was causing the breathing problems. The animation ends with a banquet in the troll village. Everybody is invited.

2.3.4 Visual style

The graphics were made by Margaret Plaisted and my role in that was very minimal. I merely gave suggestions or helped with brainstorming.

The illustration style was designed to be visually rich and detailed, which is consistent with the research Margaret and Suvi Kitunen carried out into children's illustration styles and technique preferences. They found that young children were most engaged with detailed illustrations such as the styles found in *Where's Wally* or *Mauri Kunnas* books. The style is bright, colourful and can be defined as "simplified realism".

The main character and key game characters were outlined so they stood out from the background, and all background illustrations were neither outlined or stroked. Characters were optimised for animation in Adobe Flash, with each limb separated at the joint, exported from Adobe Illustrator CS4 as a SWF file and made into individual movie clip symbols. This process was later changed.

Game scene illustrations were designed so that they could be loaded into the game piece by piece rather than loading the whole scene at once. Each illustration piece had to align perfectly for this to work. I will talk more about this in chapter 3.3.2.

2.3.5 Sound vs. no sound

"Ah, what would a rhythm game be without sound? Answer: Boring."

– Pinky Street game review at GameFAQs[9]

The question of using sound in the game resulted in a heated discussion within the team. The main issue was about treating the players equally. If the rhythm was audible and players could hear, they

9 Http://www.gamefaqs.com/portable/ds/review/R121213.html

would have an advantage over other players who can not hear. We did not intend to create a game that gave players unfair advantages over others, instead; a game that can still be played competitively.

On the other side, the sounds in the games usually enrich the experience profoundly. As a hearing player I would not enjoy the games as much if I was playing them without sound. This is maybe partly because I am used to have the sounds in the games. I enjoyed the games I was playing as a child as much or even more than the games today even though the sounds used back then were much lower quality and minimalistic compared to today's standards.

The discussion often became personal between the hard of hearing team members, sometimes it was so intense that it caused unnecessary tension in the team. This might have been caused by some more fundamental issues. In any case, this episode left a mark in the later relations of the team members.

We had some ideas for trying to solve the issue, though. One idea was to try to use sounds that could be felt as vibration by a non-hearing person. However, we soon realised it would require speakers with better sound quality than an average computer speaker. We wanted our game to be accessible to people without special equipment. I also made some sound tests with Adobe Flash CS3 and found out that playing the sound rhythmically affected the rhythm accuracy so much that it became unusable for our purposes anyway. The program slowed down slightly while playing a sound.

In the end we ruled out using the sounds as rhythm feedback. We still left open an idea to use some ambient sounds for the general atmosphere of the game.

Whenever I have been testing our game, I have mentally placed imaginary sounds as rhythm feedback. I do not any more feel the need to have the computer play them. It turned out that both the non-hearing and hearing children have seemed to be happy to play it without any sounds. Thus, I would not entirely agree with the quote about the rhythm games without sound any more. However, I know our game would not be as engaging for hearing adults as it is for out target group. This is not a fault of the game, instead it reflects our success in communicating with and engaging our target audience.

2.4 Gameplay in some rhythm games

During the initial phases of the project, we gathered information on rhythm games in general. To tie

our game into the context of gameplay with some existing games I will give a few examples.

Guitar Hero and Rock Band series

Both *Guitar Hero* (Activision, 2005-2009) and *Rock Band* (Harmonix Music System, 2007-2009) games are played with mock-up music instrument game controllers. Both games have music as one of their key element, and the beats are based on the rhythm of the song.

The beat is visualised as colour tabs which slide vertically from top to bottom (Figure 1). When they reach a certain point on the screen, the players are supposed to act with the instrument peripheral they are using. For example, if players are using a guitar peripheral, they need to press buttons corresponding to the colour of the marker. With a drum peripheral they need to strike a drum head with the same colour.

Figure 1: Guitar Hero 5 (Activision, 2010)

Dance Dance Revolution (DDR) and its clones

Dance Dance Revolution (Konami, 1998 - 2010) is a series of dance games played with a dance pad peripheral. The dance pad has usually a 3x3 matrix of square panels that can detect the player's steps on them.

The player is supposed to step on the pad in a rhythm defined by flying arrows on the screen (Figure 2). The pad has corresponding arrows to step on.

Many similar games have been made after *DDR* and there are many ways to control them, for example keyboard.

Figure 2: Dance Dance Revolution X (Konami, 2008)

Wii Fit's Basic Step

The game is part of *Wii Fit* (Nintendo, 2008) game collection, and is played with Wii Balance Board. In the game, the player has to step on and off the Balance Board in a certain rhythm. The rhythm is visualised as rectangles with foot prints scrolling vertically from top to bottom (Figure 3). The foot prints on the screen guide the stepping. The player has to follow a framed rectangle and place the feet according to the foot prints inside the frame.

Figure 3: Wii Fit Basic Step (IGN, 2008)

Audilex

Audilex (CompAid, 1998) is a game developed by Kai Karma (1998) as part of his research on auditory dyslexia.

There are two parts in the game. In the first one the computer draws two random series of images on the screen (Figure 4). After that it plays a melody corresponding to the other image series. The image series and the sound move from left to right, like in reading English. The player is supposed to select the image series that corresponds the melody heard.
In the second part of the game the computer draws an image series and plays a corresponding melody or rhythm. The player has to press a button when the last image is reached.

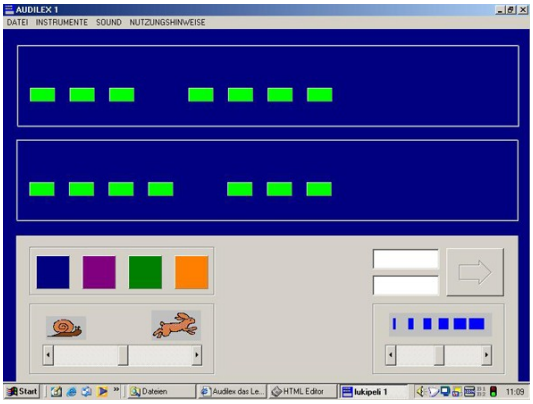

Figure 4: Audilex (Cognaid, 2010)

Restore rhythm

Restore rhythm (Mikhailov, n.d.) is a simple online game. In it the computer plays a rhythm. The player's goal is to repeat the rhythm with the help of visual and auditory cues (Figure 5).

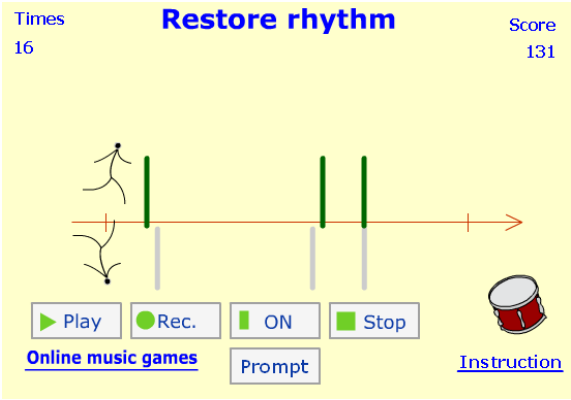

Figure 5: Restore rhythm (Mikhailov, n.d.)

All aforementioned games use sound as one element of the rhythm. It is used either as a cue or feedback. In some cases it is used for both purposes.

3 Developing The Brave Little Troll

In this chapter I intend to explain the technical solutions and the structure of the game. I will also talk about the development process on a general level.

3.1 Examining technical possibilities

In the early phases of the project we charted out technical possibilities for the game. I will go through them here.

Adobe Flash

We decided quite soon that the game would be developed with Adobe Flash. I had experience in programming with ActionScript 2, the language used in Adobe Flash environment. I was also in the middle of learning ActionScript 3, the new version of it. We also wanted to have the game playable online as one option, so this limited our choices. Adobe Flash seemed to be the best[10] one at that time. It had the best browser penetration[11] at the time.

After deciding on the developing environment we started to map out different possibilities for playing the game. We were brainstorming about the different ways of producing rhythm that could be used with a computer game. We then limited the possibilities to widely available solutions, so the audience could easily get access to the game. We ruled out custom made gadgets with certain sensors which could otherwise have been used.

Webcam

Among the first solutions we tried out were a webcam and a microphone. Teemu Korpilahti and Suvi Kitunen made a demo with a webcam and movement tracking. However, that did not work well enough, because the movement tracking was not very accurate. It was affected by the environmental variables like room lighting, the background behind the tracked person and the colour of the clothes the person was wearing. In addition fast movements were not detected. The image quality differs quite much in different webcams. Some are more sensitive to light than others. These issues made us decide against using a webcam as a game controller.

10 Panda3D seems interesting environment. It is free, efficient, seems to be reliable and the games can be programmed with Python which is relatively easy language to learn. The games can be created for multiple platforms and for web as well. Unity is another powerful game developing environment. It uses .NET-based scripting. There is a free license version of it with web browser integration as well.

11 According to Adobe, Flash reaches 99% of the Internet users. http://www.adobe.com/products/player_census/flashplayer/

Microphone

Teemu Korpilahti made a test with a microphone. We had ideas of clapping hands rhythmically or blowing rhythmically to the microphone. There were again issues with the environmental variables, such as background noise, the loudness and duration of the sound produced by the player, and different sensitivities of microphones. The accuracy of the sound detection was not good enough for our purposes and we decided not to use that either.

Wii Balance Board

We wanted the body movement into the game in some form. Nintendo Wii has peripherals that are easily available and have many of the features we were having in mind. Wii Balance Board can be used for detecting the movement of the body. It is a board with several sensors which can detect weight, and are quite accurate. The board detects even small changes in body posture while standing on it.

Wii Remote and the Balance Board can be paired with a computer using a Bluetooth connection. Some software is then needed to receive the data from the controllers and to relay it to Adobe Flash for example.

I found a few projects that had been involved in hacking Wii peripherals for using with a computer. There was already a lot of information about hacking Wii Remote, but the Balance Board was so new device, that there was not too much information about that. Some projects started to appear during the project and finally I found one suitable for our purposes.

WiiFlash[12] is a project which has provided means to use the Wii controllers with Flash. It consists of a server and an application programming interface (API). The Wii Balance Board had been supported by both Windows and Mac since version 0.45.

Wii Remote

The Wii Remote has several different sensors that enable it to detect rotation and movement. For example: it can be used for detecting arm movement when held in one's hand. It also has a vibration feature, so it can be used for producing tactile feedback.

The keyboard and the mouse

12 Http://wiiflash.bytearray.org/

The keyboard and the mouse are the most common input devices for home computers. Thus they were our first choice for the default game controllers. They can not be used for detecting body movement, though. They can be used for detecting finger tapping or hand movement. In *The Brave Little Troll* we detect the finger tapping.

Timer in Adobe Flash

We thought of using a software timer as a tool for either timing visual feedback or to measure the accuracy of rhythm the player was producing. The timer in Flash, however, turned out to be inaccurate. This finding got support from some blogs dealing with ActionScript. Since the time I was making the tests, the situation may have changed.

3.2 Graphical user interface

In this chapter I aim to describe the game menus and the game interface. I also aim to explain some design decisions and challenges related to them.

3.2.1 Main menu

The first version of the main menu had only one button. It was for starting the game. The final version of the menu is a result of two major iterations.

First we added a button for accessing the settings menu. The reasons for adding the settings menu is explained in the chapter 3.2.2.

The second major change was the addition of a difficulty level button.

Difficulty level

We began to think about adding varying difficulty to the game when we felt the game was too monotonous and boring. We felt that the same action was repeated over and over with the regular speed with no surprises.

We decided to make the game slightly more challenging by adjusting the tempo of the game. Initially we thought of increasing the tempo as the player progressed in the game. Later on we decided it may be better to make the game adapt to the player's skills by adjusting the difficulty based on the player's performance. The tempo increased when the player made perfect hits certain number in a row and returned back to normal with a failed hit or a “good” hit.

I consulted with a rhythm teacher Elina Kivelä-Taskinen. According to her decreasing tempo brings more difficulty to the rhythm producing than increasing it. I decided to modify the difficulty adjusting so that instead of only increasing the tempo, it either increases or decreases randomly within certain limits.

Many games have this kind of dynamic difficulty adjusting built in. The benefits of balancing the game this way have been discussed by Salen & Zimmerman (2003, pp.212-228).

During a game test session[13] I noticed that the adjusting tempo produced more difficulties than expected. There was no visual feedback yet about the changing tempo. It simply changed, and the player only noticed this as a slower or faster animation. It was interpreted more as a glitch than a feature of the game. Playtesting revealed the gameplay too frustrating in some cases, because the changes in tempo came as a surprise. I decided to add arrow animations to indicate the change. I also decided to give players the option to choose whether they use the adjustment feature or not. Hence, there is now an option to choose an “easy” difficulty level with the regular tempo and a “hard” level with the adjusting tempo (Figure 6).

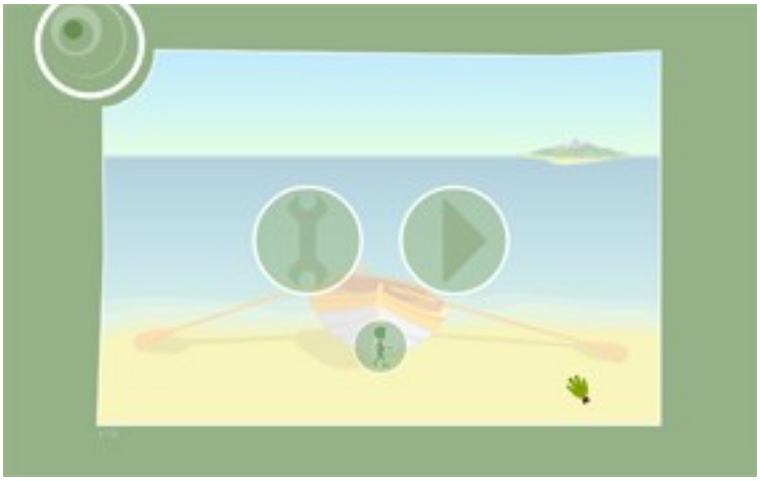

Figure 6: Final version of the main menu

3.2.2 Settings menu

The settings menu was created after we had a meeting at the Finnish Association of the Deaf. In the earlier versions the interface colour had been orange. As the interface developed, Margaret Plaisted changed the colour to green so that the user's attention was directed at the content rather than distracted by the bright orange interface. This also brought the game into line with the overall look and branding style of the project. In the meeting, however, one deaf participant pointed out she had

13 We had one game test session with two deaf children. The other one belonged to the target group. He was 7 years old. The other one was 10 years old. A plan for the test can be read in appendix I. We did not manage to arrange more tests with children belonging to the target group.

difficulties in focusing on the rhythm, because the difference in the colours between the level graphics and the interface was not as big as before. Especially in the river level the background seemed to blend into the interface. She explained she experienced some kind of motion sickness because of this. She said the reason to this is probably her dyslexia. This was an important remark, as dyslexia and sense of rhythm are linked. During the discussions with people from the Deaf Association we also found out that the Deaf are in general more sensitive to visual noise than hearing persons. They get easily distracted by it.

We did not do tests with different colours, but we did not want to change the colour back to orange, because many people had told us the green looked better. However, the interface colour seemed to be a problem and we decided to do something about it. First we thought of having a colour cycle which would have allowed players to choose any colour they wanted. This would have required a total redesign of the interface. It would also have been a very difficult job to re-colour all the different elements and still maintain the style. It would have been very difficult to automatically find the tones or colours that go well together.

We ended up having two possible colours to choose from. The green was left as the default colour and the original orange as an optional one.

In the first version of the settings menu, the colours were changed with two separate buttons (Figure 7), but that seemed to cause confusion during the game test. A child was asked to change the colour and he managed to do it, but after that was not able to change it back when he wanted to do it.

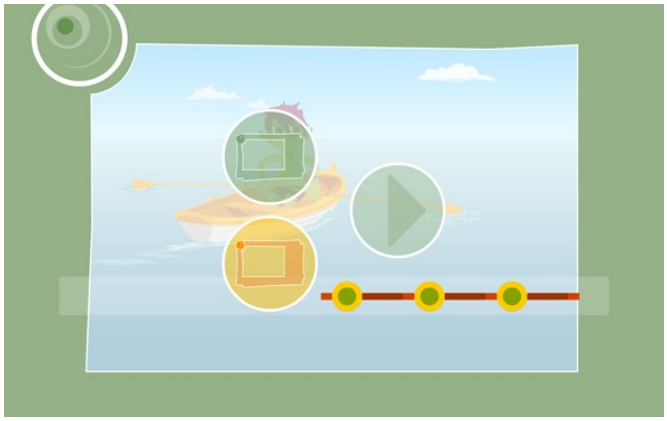

Figure 7: Development version of the settings menu

In the final version the colour button is a toggle which switches the colour (Figure 8). The colour of the button itself changes to the opposite colour. At the time of writing, the last version has not been

tested with the target group.

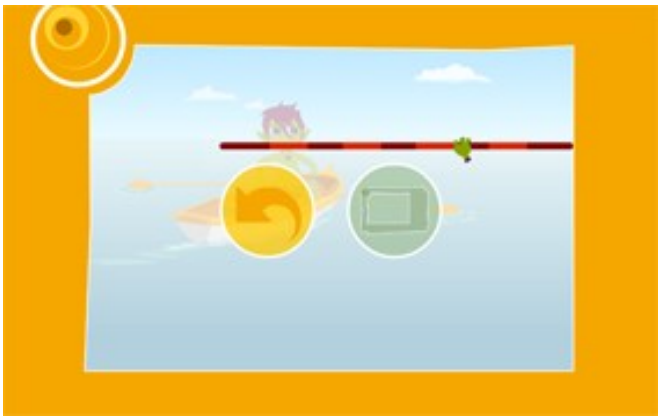

Figure 8: The final version of the settings menu

Another issue that was brought up in the same meeting was the position of the rhythm bar. It seemed people had different preferences. Some people said it was easier to focus on the rhythm bar when it was closer to the troll. Others said it was easier to focus on it when it was on the bottom of the screen. I felt the best option is to let the player choose the position dynamically within certain limits. The limits are important for keeping the bar not getting into an unsuitable location. The rhythm bar can now be dragged to a desired position.

The menus can be navigated both with the mouse and the keyboard. The arrow keys are used for navigation and the space bar and enter key for making the selections.

3.2.3 Game interface

Scoring

We wanted to have a scoring system for the game, so the players could see how well they performed. We used numerical scoring in the development versions, but it did not seem the correct way considering the target group. Children of that age do not necessarily understand the relation of numerical values which we used (from zero to few hundred).

In the final version we decided to use a visual indicator to display the score. We used colour coded bars for each level. The bars grow longer with successful hits and diminish with failed ones. With a perfect hit on the green area of the rhythm bar the score bar grows slightly larger than with a hit on the yellow area.

The final result is not the most intuitive one and some players do not even notice it. If they do, they sometimes think the bar indicates their progress in the game. In that sense our design was a failure. We ran out of time, however, and this was the best we could achieve. It could be redesigned in the future. One possibility would be to use smaller numerical values for the total score. The score would grow slowly, for example one point per a certain number of successful hits.

We also had an idea to create high score table, so the players could compare their performance with themselves or the other players. This would have required some form of player setup and that would have added complexity in the overall game experience. Some players also find competition with other players discouraging as Schell (2008, pp.105) points out. In the end we decided not to record the scores.

Level progress indicator

At first we did not have any indicator for the level progress. Based on the feedback we got from people who were testing the early versions of the game we realised the indicator would be quite useful. Playing the game without knowing how long the level is seemed to cause slight anxiety.

The progress indicator went through a couple of visual iterations. At first it portrayed the progress of the level with a bar and two troll faces. The other face had a sad expression and the other one was smiling. The further the player progressed the more visible the happy face became. The sad face faded away at the same pace.

The final version was Margaret Plaisted's idea. It depicts a path between the starting and ending points of the level.

The visuals of the new indicator could have been tested more. At the time of writing this I have heard claims the indicator is not visible enough or the meaning of it is not easily understood. However, these claims have not been made by any members of the target group.

Life symbols

The final version of the game interface has three troll heads which represent the lives of the game character. They also show the health of the character. With every failure the troll's health suffers. A life is lost after three failures.

I might modify the health system in a future version to be connected with the player's performance. Good performance could be rewarded with health or life regeneration. However, testing should be

carried out to ensure this would not disrupt the balance of game play too much.

'Reset' and 'home'

The reset button takes the player to the beginning of the level. The home button takes one to the beginning of the game.

In the version 1.01 both buttons are only available during the level played. The 'home' button should probably be placed into the main menu instead. Now accessing it is too difficult, because one has to start the level first. I will probably change this in a future version.

There seems to be similar problem with these buttons as with the progress indicator. They are not necessarily either noticed or understood as buttons. In a possible future version, these issues should be addressed and solved.

3.2.4 Rhythm bar

The rhythm bar is a central element in the game. It gives the player a visual cue for the rhythm that is followed. The bar went through several different versions before it found its final form.

In the early phase of the project I made a simple rhythm game prototype in which one could create a rhythm pattern moving coloured circles on a line (Figure 9). When the game was started, a marker began moving on the line. When the marker hit a circle, one had to press a key to get a score.

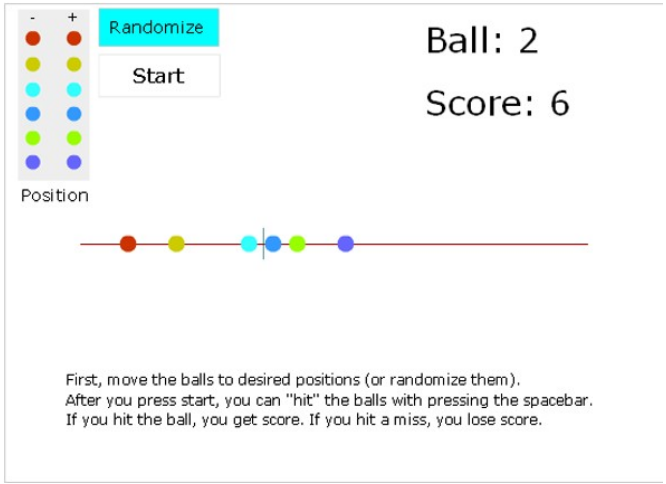

Figure 9: An early rhythm test

When the game started to take shape, an initial rhythm bar was tested. It had colour areas for different hit values: fail, good and perfect. A marker moved through the bar, and player had to press a key when the marker was on the desired spot. The bar, and the time the marker went through it, grew longer or got shorter depending on the rhythm data. (Figure 10).

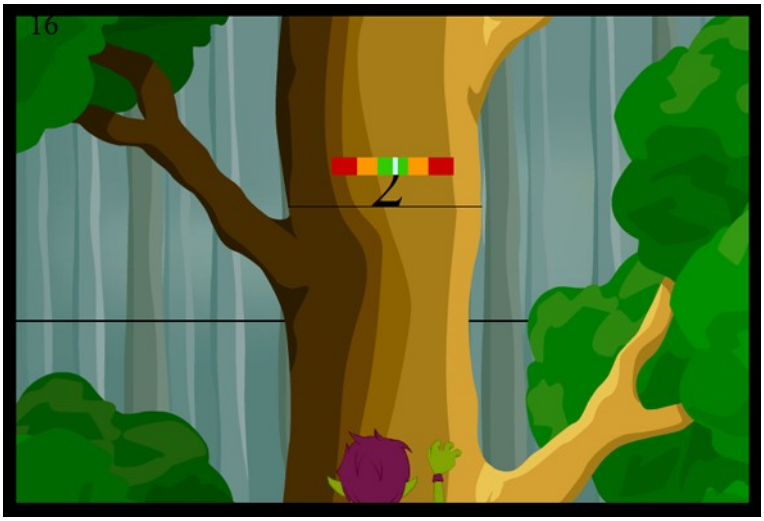

Figure 10: First test version of the rhythm bar

This solution did not provide the feel of rhythm well enough. It was also very difficult to anticipate how the rhythm was going to change. The only cues were in the tree graphics, because the tree branches provided the rhythm. However, these could not be perceived further than one or two steps ahead which was also a problem.

The second test version of the rhythm visualisation resembled slightly Wii Fit's Basic Step mentioned in chapter 2.4. In it a row of figures were moving from right to left. One by one they entered a square which marked the active beat. The figure filled gradually with colour and the fill position marked again the hit values. This worked slightly better than the previous test version. However, it was still difficult to perceive rhythm with it. (Figure 11).

Figure 11: Second test version of the rhythm bar

The final version of the bar (Figure 12) was a refined version of a slightly similar bar we used in the first playable demo of the game. I will talk more about the demo in chapter 3.3.2.

Figure 12: Final version of the rhythm bar

The rhythm bar consists of lines and circles. The lines represent pauses and the circles represent beats. A static marker that can be seen in some later images of the game, is used to represent the hit area. The same colour coding is used as before. Hit on the red area means failure, yellow area is a good hit, and green area is a perfect hit. The bar moves either horizontally or vertically depending on the level.

With this version the rhythm is easier to perceive and anticipate. If the bar is compared with the first rhythm prototype, one can see that the form of the final version is quite close to that again. However, the beats are fixed, and the bar moves instead of the marker.

3.3 Level design

To give a general idea on how the game works, I will describe the structure of the game levels.

3.3.1 General level structure

In order to streamline the development process Sven Noben proposed an approach that became the basis of the level structure. It was modified through several iterations, and between the first and final version only the idea is similar. The approach was to construct the levels using XML data which contained the rhythm data and the data defining the level graphics separately.

All the game levels except the tutorial level are constructed this way. The level graphics are modular and bound to the rhythm pattern that is constructed with the XML data (Text 1 and Figure

13). This enables very easy modification of the game levels.

```
<root>
      <pattern>
            <bar>4</bar>
            <bar>3</bar>
            <bar>1</bar>
            <bar>3</bar>
      </pattern>
```

Text 1: The first part of the river level XML data

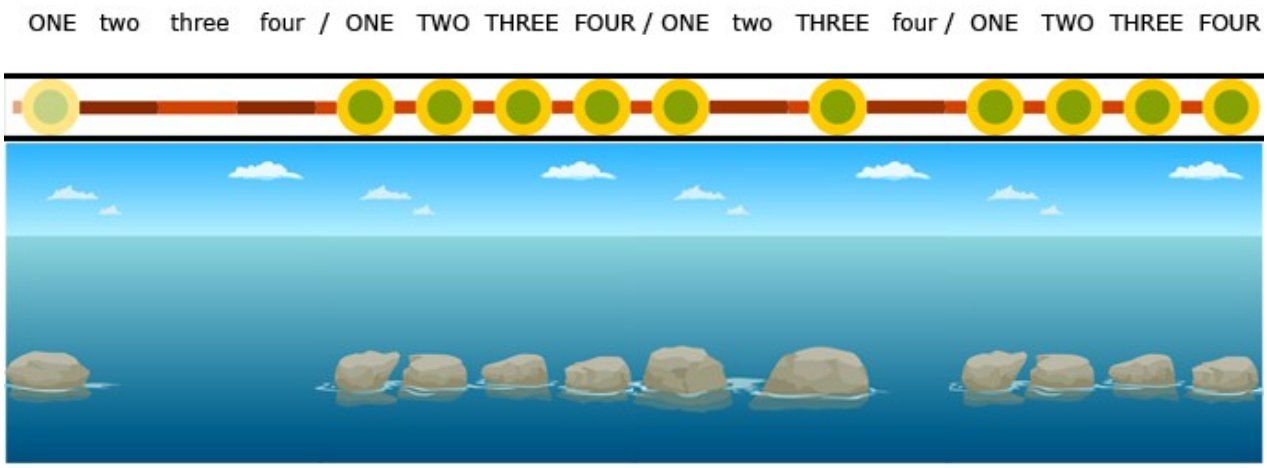

Figure 13: The river level graphics based on the XML example

The rhythm pattern can be constructed with up to three different elements. These elements can be put into any order and basically the total rhythm pattern does not have a limit. For the purposes of the rhythm training, simple rhythm repetitions are used. Longer and more complex patterns would be possible to create as well.

The elements we are using are quite limited. Only combinations of **ONE TWO THREE FOUR** / **ONE two THREE four** / **ONE two three four**[14] can be created, but not for example **ONE TWO THREE four** or **ONE two THREE FOUR**. This makes the compositions one can make rather simple in general.

This was a design decision we made quite early in the project. Our knowledge and understanding of the rhythmic compositions was limited and in retrospect we limited our future options too much with the decision we made. When we realised how limited our compositions were, it had become impossible to add new kind of rhythm elements to the game without building a new game engine

14 The beat is on the capitalised words, like 'ONE'.

and rethinking the whole game structure with animations and gameplay again. We simply had to accept our earlier decision and make the best we could with it.

If we had been able to create more complex and interesting compositions, the game might have been more interesting to a broader audience as well. The children have seemed to enjoy the game, but I have not been able to test or follow how quickly they get bored with it.

One problem which was pointed out to me after the game was published is that in the end of the level the rhythm pattern may end in the middle of a rhythm. It had not been noticed by the test players before and I had become blind to it during the project.

The rest of the XML has definitions for the troll animations and the default frame rate of the level (Text 2). The frame rate defines the tempo.

```
      <definitions>
            <animation>
                  <jump>3</jump>
            </animation>
            <animation>
                  <jump>1</jump>
            </animation>
            <animation>
                  <jump>0</jump>
            </animation>
      </definitions>
      <animspeed>
            <frate>24</frate>
      </animspeed>
</root>
```

Text 2: Definitions for the troll animations and the default frame rate.

For each level, after the tutorial, there are four XML files from which one is loaded randomly. As a result, there are 256 different possible combinations for the whole game. This produces more variation for the game rhythm-wise.

3.3.2 Tree level

This was the first level created for the game, so it was the most challenging level to build. It

provided a model for the other levels when it was in its final stages.

Before the approach using XML data in level creation, a cruder approach was tried. We had an idea to use a palm tree into which the troll would climb after a monkey. The troll would “slide” across the trunk and the player would have to hit left or right arrow keys when the troll was in certain position. The rhythm would be produced by these hit locations (Figure 14).

The palm tree consisted of one very large piece of graphic. This approach proved to be be problematic. The first problem was the size of the graphic. In the beginning I was working on Adobe Flash CS3. The size limits in it for the graphics are stricter than in CS4. This meant the tree could not be as long as desired. The length caused functional problems as well.

The rhythm markers were not following the shape of the tree. I tried to solve this by using a collision detection algorithm to attach the markers to the trunk. However, this was computationally too heavy task. The program either froze during the initial setup when the markers were being added, or playing the animation was too slow. This happened even with powerful computers.

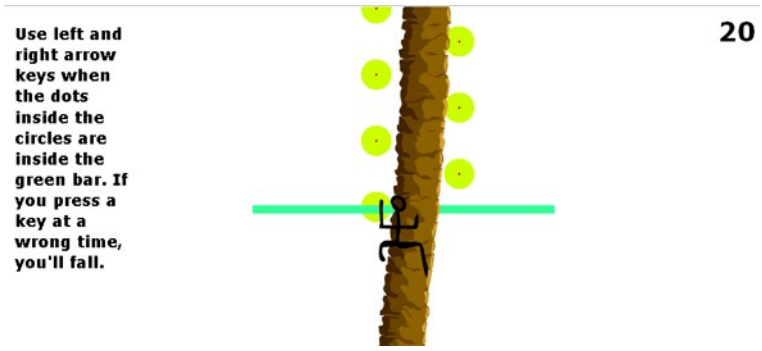

Figure 14: Early prototype of the tree level

These problems were the main reason we started working based on Sven Noben's approach. After that had been developed into its final form, the tree was divided into sections. These sections are connected to each other like film frames. The sections are moved in a row. There is always one section placed outside of the screen to wait for its turn. The sections are removed when they move out of the screen again. (Figure 15).

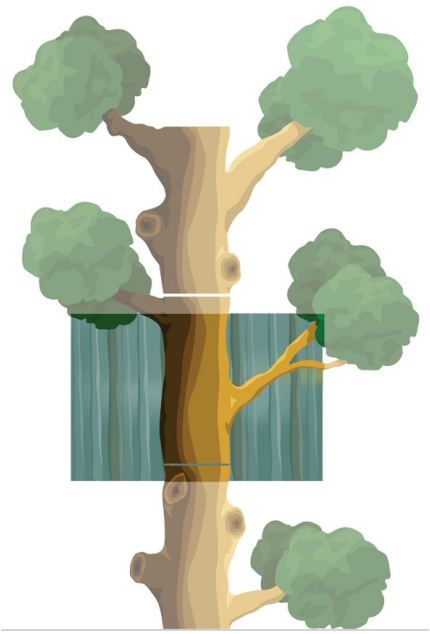

Figure 15: Divided sections of the tree animation

Iterations

The first playable version of the level had already many of the features of the final version. It was put to test during the Media Lab's Christmas Demo Day 2008[15]. During the project we had different stages of the game displayed and playable in several public events. Some of the players were children and some of them were adults. We were observing the players and took notice of the problems that occurred. This was our main method for play testing. The events where the game was playable are listed in appendix II.

We were using a separate rhythm bar instead of hit locations attached to the tree graphic. The rhythm bar was located on the bottom of the screen in horizontal position. Because the level graphics were moving vertically and the rhythm bar was located far away from the troll, the players had difficulties in following the rhythm.

Another problem related to the graphics was the troll's movement range. If the player failed a couple of times, the troll could slide to the bottom of the screen. There was a safety limit, so the troll could not fall outside of the game area. However, if the troll fell to the bottom, it most often also stayed there (Figure 16). It got up temporarily during the jumps, but fell back after them. This was problematic both visually and gameplay-wise.

15 Media Lab Demo Day is twice a year organised event where the students and research groups of the Media Lab Helsinki present the projects they have been working on.

Figure 16: First playable version of the tree level. The troll is on the bottom of the screen.

In the final version the rhythm bar is aligned vertically with the tree graphics. Its default position is on the side of the troll (Figure 17). The troll does not fall on the bottom any more, because it makes a jump even if the player fails with the rhythm. The failed jump is clumsier than a successful one, and the animation aims to show the troll struggling not to fall.

Figure 17: Final version of the tree level

The game flow of all levels except the tutorial level is similar to the tree level (Figure 18).

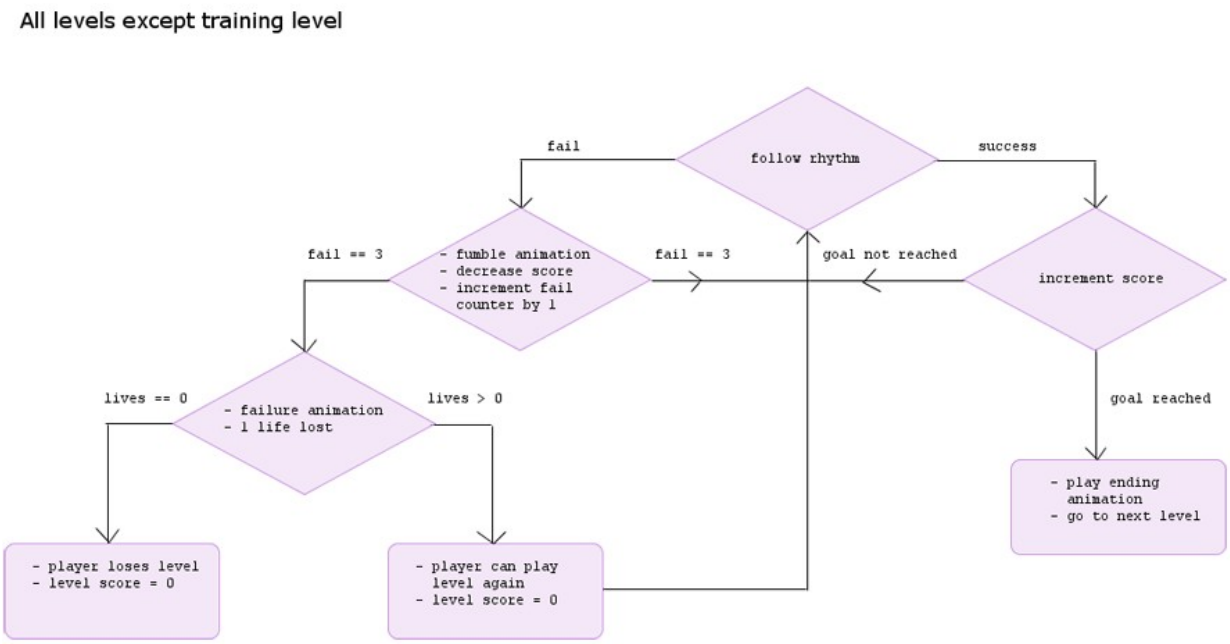

Figure 18: Game level flow

The rhythm elements used in the tree level are: **ONE TWO THREE FOUR** and **ONE two THREE four**.

3.3.3 River level

On this level the troll is crossing a river by jumping on rocks. The level scrolls horizontally from right to left (Figure 19).

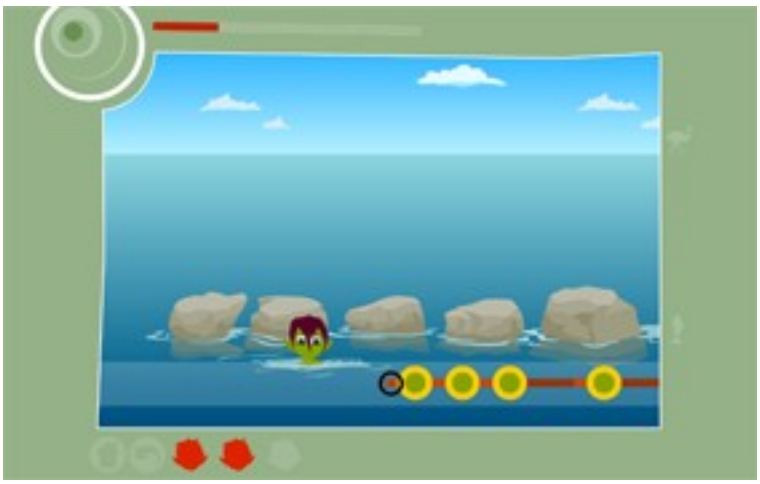

Figure 19: Final version of the river level

Building this level had its own challenges, because the scrolling is now horizontal. The

functionality of the tree level provided a basis for this level, but it had to be modified quite heavily.

The rhythm pattern is constructed with all three possible elements: **ONE TWO THREE FOUR**, **ONE two THREE four** and **ONE two three four**.

One of the most problematic issues on this level was the troll animation's relation to the background movement. The way the troll was animated, and the way the animations were implemented in the programming, resulted sometimes the troll levitating on the water instead of landing on a rock. To solve this issue I used two tricks. The simpler one was to increase the size of the rocks, so the troll would always land on them. Another trick was to add code to make the jumps gradually shorter if needed.

There exists still another smaller problem in the troll animation. When the troll starts the jump, he sometimes makes a small jump backwards. This is caused by the way the animation is made. I have not been able to solve the problem neatly enough by the version 1.01.

3.3.4 Mountain path level

On this level the troll is riding an emu on a mountain path towards the castle. The path has a lot of gaps, so the emu has to jump over them. The level scrolls horizontally like the previous level (Figure 20).

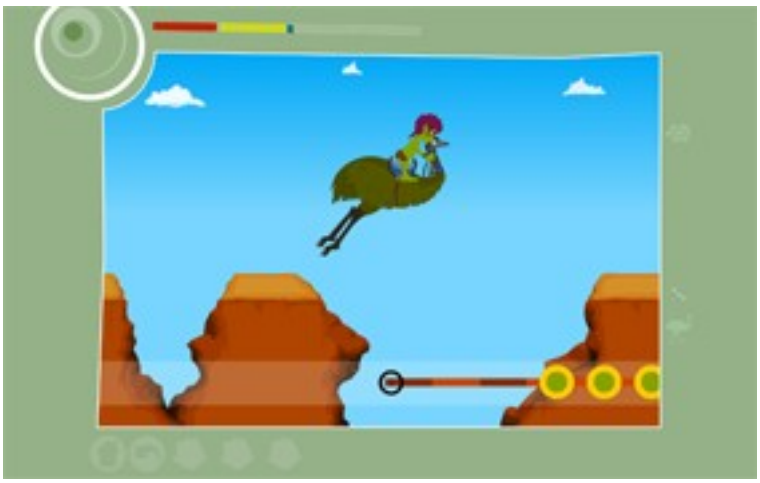

Figure 20: Final version of the path level

The rhythm pattern is built with the same elements as in the previous level, but it is intended to be slightly more complex.

Previous level's jumping animation problems existed on this level as well.

3.3.5 Castle wall level

On this level the troll is climbing up the castle wall with the help of special gloves and boots. The troll jumps diagonally from bricks to bricks. The level scrolls vertically (Figure 21).

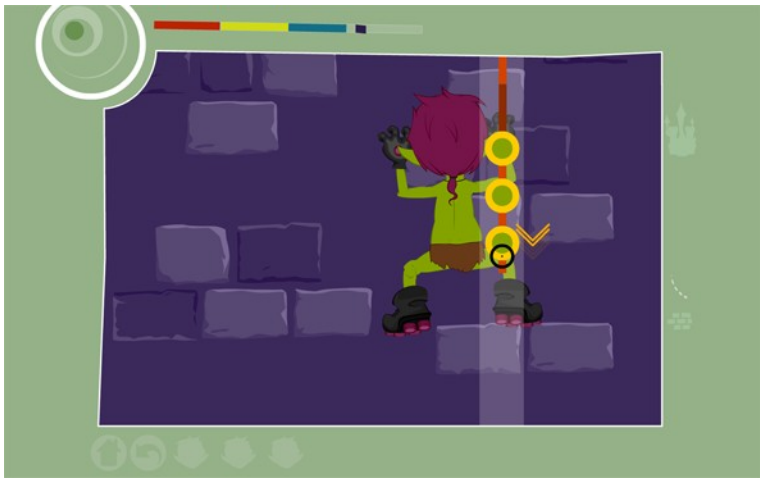

Figure 21: Final version of the castle level

The rhythm pattern is constructed again with the same elements as previous two levels. It is again intended to be slightly more complex.

For some players this level has been a bit more difficult than the other levels. The troll's diagonal movement might be the reason. In the other levels the movement of the troll has been aligned with the movement of the rhythm bar. The movement is slightly more distracting, probably because it deviates from the familiar pattern, and the rhythm pattern is not aligned with the troll any more all the time.

3.3.6 Tutorial level

This level is the very first level in the game, but it was built last. It is in many ways different from the other levels. It was designed to teach the player how to play the game.

On this level the troll is rowing a boat. With successful hits the troll rows. With failure the troll drops the oars (Figure 22).

Figure 22: Final version of the tutorial level

The rhythm pattern is constructed with only one element, **ONE two THREE four**, so it is very simple. There is an animated hand pressing the space bar at the same time as the player is supposed to press it. The test players made the connection usually very quickly.

The level progression differs slightly from the other levels. Instead of continuous progression, the player has to make a certain amount of successful hits in a row. If the player fails, the progress indicator returns to the starting position and the calculation starts from zero. There is no effect on health on this level and one can not “die”. There is no score counted either. (Figure 23).

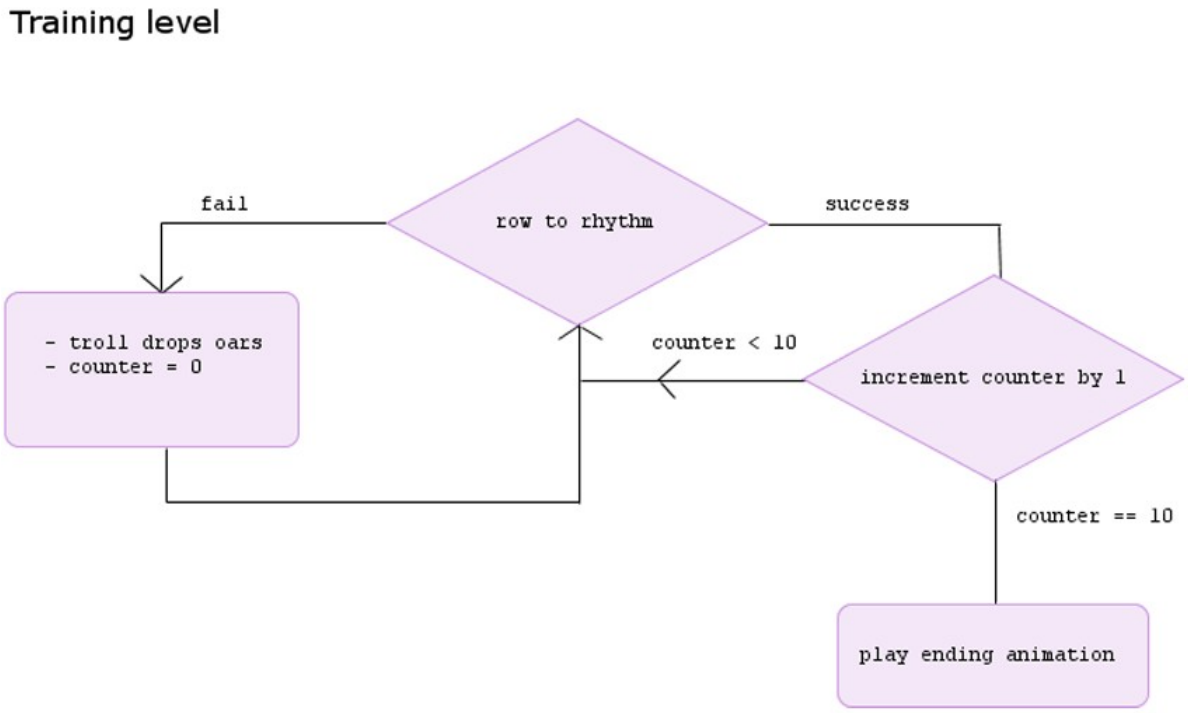

Figure 23: Training level flow

4 Finalising

Finalising the game included optimising the performance of the game, polishing the functionality and graphics, and creating a web version of the game. In this chapter I will go through these phases.

4.1 Optimisation

Optimising the game for better performance happened during many stages of the development. There were for example issues with the game speed that slowed down both due to memory leaks and less optimal programming solutions.

The memory leaks happened mostly during the development of the first level. The game slowed down to the point of not being playable at some point. The reason for this was difficult to point out until I started to suspect memory problems. I found a SWF profiler[16] which helped me to find both the reason and the cause of the problem. Certain elements in the program multiplied instead of being removed. Removal should have happened at the end of the level. More technically, some event listeners handling the level animations continued their life in the background while new event listeners were added when the level started again. The solution was simply to forcefully remove the event listeners each time the animations were removed.

There are many ways to perform tasks programmatically. Some of them are more efficient and faster than others. Finding optimal solutions is important when dealing with tasks that require lots of processing power. Moving large graphics is this kind of task. I consulted a few blogs[17] for finding optimal solutions for our purposes. Some of the most useful hints for my purposes were dealing with faster array iterations and math calculations.

The graphics can be optimised with the tools in Adobe Flash environment. This can be done both manually and automatically. The tools are not very efficient. Automatic optimisation causes quality loss easily. Manual optimisation takes a lot of time and is not necessarily very efficient.

4.2 Polishing

Recolouring

16 SWF profiler was made by Shane McCartney. It monitors the frame rate and memory use of the SWF file.
http://www.lostinactionscript.com/blog/index.php/2008/10/06/as3-swf-profiler/

17 Open Source Flash: http://osflash.org/as3_speed_optimizations
Nick Bilyk's blog: http://www.nbilyk.com/optimizing-actionscript-3
Grant Skinner's blog: http://www.gskinner.com/blog/archives/2006/08/as3_resource_ma_2.html

Margaret Plaisted was originally creating the graphics in Adobe Illustrator CS3. During that time there were no problems. In January 2009 she upgraded to CS4. She exported SWF files for Teemu Korpilahti to animate. In the process of exporting from CS4 and Teemu bringing them into CS3, the tones changed slightly. The colours were washed out and dull compared to the original files. We did not notice this for a long time and so a large percentage of the graphics had a wrong colour tone when Teemu and I were working on them.

When Margaret noticed the problem, the damage was already huge. Many of the animations were made frame based and recolouring the graphics in hundreds or thousands of frames seemed to be a very tedious job as we thought it would have to be done manually.

I had already recoloured some of the shorter animations manually when I suddenly got a feeling there must be a smarter way to do it. I had concentrated only on the programming part of the Adobe Flash CS4 environment and had neglected almost all design features. After some searching I found out that there really is a tool for searching and replacing colours automatically in all the frames. Instead of us manually changing the colours wasting hours with a dull task, the computer changed the colours in a few seconds. This made me realise how important it is to familiarise oneself with the tools one is working with.

Debugging

During the project I realised the importance of working efficiently and making sure every step is documented and the code structure is diligently commented. My code commenting style is minimal if there is any and I tend to keep the structure of the program in my memory instead of making a UML model or diagram of it. With smaller projects it is not so much of a problem, but for larger projects it has more impact. As the project continued over a long time, and there were sometimes long breaks in my work, some bad problems in the code became difficult to solve. They were simply too well hidden in the slightly chaotic structure.

When I had been desperately hunting the bugs for a while, I realised I had to make a model of the existing structure. It took some time to create, but after that it was relatively easy to find the location of bugs during the rest of the project (Figure 24).

Figure 24: The structure of the program at some point of the development

4.3 Web version

4.3.1 Loading animation / preloader

The game has both shorter and longer animations between the levels. In the web environment loading them may take relatively long time depending on the speed of the connection. Loading the game itself takes also quite a long time as the main file is large. Showing an animation during the loading seemed to be a necessity. A preloader is such an animation. It usually shows how much of the data is loaded and how much is to be loaded.

I took the task of designing the animation. I was fortunate to have a large pool of game art Margaret Plaisted had produced. It was quite clear to me from the start that the animation should somehow be related to the game. I spent almost a day testing different solutions and thinking about different versions. The final version was a result some sudden spark of intuition. It had some elements of the versions I had made, but the final form simply appeared in my mind, and immediately felt the right one.

The preloader worked fine when the main file of the game was loading, but caused severe problems when the animations between levels were loading. I was not able to find the cause for the problems, but realised I can use a cheap trick. I did not need the functionality of the preloader in those cases. I just needed an animation to play during the loading times. So, I just inserted the animation there.

During the loading time the orb hearts around the troll's face become more visible one by one

(Figure 25). The troll's eyes follow the events. As a preloader it shows the amount of loaded data as more visible orb hearts and the rest of it as less visible ones.

Figure 25: Preloader animation

4.3.2 Optimisation

The game gained some speed with manual optimisation, but for the web the game still had quite large animation files and complex graphics. I decided to try one commercial optimiser to finish the job. The software was Flash Optimizer from Eltima and it proved to be useful. Finding good settings for optimisation took a long time, but after I found them, the file sizes were reduced average 50 % without noticeable quality loss in the graphics.

5 Conclusions

5.1 Learning outcomes

The project was a great learning process for me. First of all working with and getting to know people from the Deaf community has been an eye-opening experience. I have also learned something about game testing with children.

My understanding about game design in general and certain game mechanics increased during the project.

I learned a lot about Adobe Flash CS4 and ActionScript 3. I am now able to create much more complex programs than I was before the project. The importance of commenting the code and keeping a design diary is also clearer to me now. I learned also a lot about hardware related issues.

5.2 The future of the game

The Wii controllers were not yet implemented in version 1.01. In a future version I will try to implement them. There are no technical constraints, but playability may provide some challenges. Usability is another issue that is challenging, because using Wii controllers with a PC and Flash games requires a special setup. This requires some tweaking both with the hardware and the software, and problems often occur.

The possible rhythm patterns are very limited in the version 1.01. In a future version the game engine could be reconstructed to make more complex rhythm compositions possible.

The game has only five levels. Some children who have played the game would have wanted more. We lacked resources for doing more, but maybe in the future the game will be developed further in this sense as well.

After the game was published, a few bugs and glitches have still been found. These should be fixed. The game interface has also some minor issues which should be solved.

5.3 Other reflections and outcomes of the project

The project and resource management was not optimal in this project. There was too much time spent on graphics related problems and resolving programming problems. There was too little time spent on game testing with the target group, and too little time spent researching on rhythm which

was the essential part of the game. Our team of three members working on the game project part-time was not the most efficient.

No research on the possible benefits of the game in developing sense of rhythm has been conducted. Thus it remains unknown whether it is a useful tool. A research is recommended, especially if the game is developed further.

During the project a paper, The Brave Little Troll: a rhythmic game for deaf and hard of hearing children, was submitted to Academic Mindtrek 2009 conference. The game was displayed there as well. The paper that I co-wrote with Suvi Kitunen, Margaret Plaisted and Päivi Rainò was later published in ACM digital library[18].

18 Http://doi.acm.org/10.1145/1621841.1621880

6 References

Benari N., 1995. *Inner Rhythm. Dance Training for the Deaf*. Routledge.

Buckleitner, W., 2008. *Like Taking Candy From a Baby: How Young Children Interact with Online Environments*. [online]. Flemington. Media Tech Foundation. Available at http://www.consumerwebwatch.org/pdfs/kidsonline.pdf [Accessed 24 February 2010].

Goswami U, (2008). *Rhythmic Timing and Dyslexia: A Causal Connection?*: Full Research Report ESRC End of Award Report, RES-000-23-0475. Swindon: ESRC. Available at http://www.esrcsocietytoday.ac.uk/esrcinfocentre/viewawardpage.aspx?awardnumber=RES-000-23-0475 [Accessed 4 March 2010].

Karma K., 1998. *Audilex, a computer program for diagnosing and ameliorating dyslexia*. Helsinki: Comp-Aid Oy.

Kitunen S., 2009. *Designing a Deaf culture specific web site*. MA Thesis. University of Art & Design Helsinki.

Kujala T., Karma K., Ceponiene R., Belitz S., Turkkila P., Tervaniemi M., Näätänen R., 2001. *Plastic neural changes and reading improvement caused by audiovisual training in reading-impaired children*. Proceedings of the National Academy of Sciences, 98, 10509-10514. Available at http://www.pnas.org/content/98/18/10509.full [Accessed 24 February 2010].

Martola T., 2006. *Rytmiä viittomiseen ja leikkeihin*. [Rhythm for signing and play]. Helsinki. Kuurojen Liitto ry:n julkaisuja 42.

Meronen A., 2004. *Viittomakielen omaksumisen yksilölliset tekijät*. [Individual differences in sign language abilities]. of Ph.D. University of Jyväskylä.

Oblinger D. & Oblinger J. eds., 2005. *Educating the Net Generation*. [e-book] Boulder, CO. Educause. Available at http://net.educause.edu/ir/library/pdf/pub7101.pdf [Accessed 4 March 2010].

Overy K., Nicolson R.I., Fawcett A.J. & Clarke E.F., 2003. *Dyslexia and Music: Measuring Musical Timing Skills*. Sheffield. John Wiley & Sons, Ltd.

Palmer, R. (n.d.). *Feeling the music philosophy. A new approach in understanding how people with sensory impairment perceive and interpret music*. [online]. Available at

http://www.kolumbus.fi/riitta.lahtinen/feeling.html [Accessed 4 March 2010].

'rhythm' (n.d.) *The American Heritage® Dictionary of the English Language,* Fourth Edition. (2003). Available at http://www.thefreedictionary.com/rhythm [Accessed 25 February 2010].

Salen K. & Zimmermann E., 2003. *Rules of Play: Game Design Fundamentals*. Cambridge. The MIT Press.

Schell J., 2008. *The Art of Game Design: A Book of Lenses*. 2nd ed. Burlington. Morgan Kaufmann.

Winkler I., Háden G., Ladinig O., Sziller I., Honing H., 2009. *Newborn infants detect the beat in music*. Proceedings of the National Academy of Sciences, 100, 11812-11815. Available at http://www.pnas.org/content/106/7/2468.full [Accessed 24 February 2010].

Computer games

Activision. (2005 - 2009). *Guitar Hero*. [DISC]. Nintendo Wii. Santa Monica: Activision Blizzard, Inc.

Harmonix Music System. (2007 - 2010). *Rock Band*. [DISC]. Playstation 2. Cambridge: Harmonix Music Systems, Inc.

Konami. (1998 - 2010). *Dance Dance Revolution*. [DISC]. Nintendo Wii. Los Angeles: Konami Digital Entertainment, Inc.

Mikhailov S., (n.d.). *Restore rhythm*. [Online]. PC. Abasoft.com. Available at http://ababasoft.com/music/restore_rhythm.html [Accessed 4 March 2010].

Nintendo. (2008). *Wii Fit*. [DISC]. Nintendo Wii. Tokyo: Nintendo.

Images

Activision, 2010 Guitar Hero 5. [image] Available at http://hub.guitarhero.com/media/images [Accessed 6 March 2010].

Cognaid, 2010 Audilex. [image] Available at http://www.opetusohjelmat.fi/kauppa/product_details.php?p=15 [Accessed 6 March 2010].

IGN, 2008 Wii Fit's Basic Step [image] Available at http://wii.ign.com/dor/objects/949581/wii-fit/images/wii-fit-20080402033458897.html [Accessed 6 March 2010].

Konami, 2008 Dance Dance Revolution X. [image] Available at http://www.ddronlinecommunity.com/static/dance-dance-revolution-x [Accessed 6 March 2010].

Mikhailov S, n.d. Restore rhythm [image] Available at http://ababasoft.com/music/restore_rhythm.html [Accessed 6 March 2010].

7 Appendix I

Game testing plan

Equipment:

- 1-2 video cameras
- 1-2 camera tripods
- permit forms
- servings
- computer and the game
- comfortable seats and a table for the children.
- notepad
- (gift)
- (Wii Balance Board)

The test:

- Welcoming and getting to know each other. Servings. Max 10 min.
- Short introduction about the computer. Starting the game.
- The child may familiarise oneself with the game in one's own pace. Video taping starts.
- A few questions/tasks about the game interface.
- Closing questionnaire
- (possibly a simple test with Wii Balance Board. Can the child control the game character with ones hands and feet in a certain rhythm.
- Short interview with the parent(s).

- End of test (gift given)

Pre questionnaire form:

- Do you play computer or video games? What kind of games do you play? What kind of games do you like?
- Do you use a computer for something else? What else do you do with it?
- Have you played rhythm games, like Guitar Hero or Dance Dance Revolution?
- Do you watch cartoons? For example?
- Do you read comics? For example?
- What kind of non-digital games do you like?

Questions and observations about the interface:

- can the player start the game by oneself
- can the player change the game settings without help (does one notice the settings menu exists, does one try to change them, and is one able to do so?)
- how long does it take to understand the game interface

Questions and observations about the game levels:

Tutorial level

- how long does it take to understand how to play the game
- is the player trying to use a mouse or does one notice the keyboard should be used instead
- does the player understand/notice the difference between success and failure

Other levels

- is the child able to play without advice
- does the child recognise the relation between pressing the key and the animation

- does the child recognise the relation between the rhythm pattern and pressing the key
- are there problems in perceiving the game events in general

Closing questionnaire:

- did the game feel easy or difficult?
- What happened in the game?
- What did you do in the game?

Interview of the parent(s):

- Thoughts about the test?
- Thoughts about the game?
- Some tips for making the test better?

8 Appendix II

The public events where the game was displayed

The following is a list of the public events where the different stages of the game were playable.

First version of the tree level

Media Lab Christmas Demo Day 2008. Media Lab Helsinki. December 18th 2008.

Premiere of *SYKE – PULSE* documentary. Light House. January 19th 2009.

Tree, river and path levels

Media Lab Spring Demo Day 2009. Lume TV-studio. May 20th 2009.

All levels

Mindtrek 2009. Hotel Rosendahl, Tampere. September 30th 2010.

Media Lab Christmas Demo Day 2009. Media Lab Helsinki. December 17th 2009.

Version 1.0

The game publication. Light House. February 12th 2010.