Audio-based game for visually impaired children

Bachelor Thesis

J.T. Coppoolse
W.J.L. van Dam

Technische Universiteit Delft

Delft University of Technology

Challenge the future

Audio-based game for visually impaired children

Bachelor Thesis

by

J.T. Coppoolse
W.J.L. van Dam

in partial fulfillment of the requirements for the degree of

Bachelor of Science
in Electrical Engineering

at Delft University of Technology,
to be defended on the 29th of June, 2015.

Supervisors:	dr. ir. R. Hendriks
	S. Khademi, M.Sc.
Thesis committee:	dr. ir. M. A. P. Pertijs
	dr. ir. R. Hendriks
	dr. ir. R. F. Wolffenbuttel

TU Delft Delft University of Technology

Abstract

This thesis was made for the Bachelor Graduation Project (Electrical Engineering). The purpose of the project was to design an audio-based game for visually impaired children. In this thesis the gameplay and the graphical user interface are designed.

We choose to make a simplified dungeon crawler in the programming language Python. We designed tutorials and levels for the game. For the tutorials the methods interaural time difference and interaural intensity difference are used to simulate localised audio. For the levels more advanced audio simulation methods are used, but these are provided by two other groups of students. A graphical user interface is made for validation purposes and for parents and caretakers of the visually impaired children.

The game is tested with visually impaired children and sighted students. The controls of the game were too complex for young children and the game was not completely accessible for the visually impaired. However, almost all test subjects were able to learn the mechanics of the game and complete levels on their own.

PREFACE

This thesis describes the design of an audio based game for visually impaired children. It is written for the Bachelor Graduation Project (Electrical Engineering). This project is done in groups of six students, which are divided in three subgroups of two students each. Each subgroup is responsible for a part of the project. In our case, the three sub-projects were: the gameplay with the graphical user interface (GUI), the audio localisation [1] and the sound reflections [2]. We were responsible for the gameplay with the GUI and we will describe this sub-project in our thesis. As the game was designed in close collaboration with the other subgroups, we will reference the other subgroups' theses throughout this thesis.

For two months we delved into audio games for visually impaired children, we researched the currently available audio games and what their capabilities and shortcomings were. With that knowledge we designed our game mechanics and GUI and tested it. But we could not have done this on our own and that is why we would like to thank everyone that contributed to the design of this game. We thank *Koninklijke Visio* [3] in Den Haag for giving feedback during the design process and for allowing us to test our game with visually impaired children. We thank our supervisors, R. Hendriks and S. Khademi, for their guidance and answers to our questions. We thank the other members of our group, M.C. Bisschop, R. Duba, B.W. Kootte and J.C. Noortman. It was a joy to work together with them in this project.

J.T. Coppoolse
W.J.L. van Dam
Delft, June 2015

Contents

1

Introduction

Background

A large portion of the youth today plays video games [4, p. 596]. Be it on a handheld device, personal computer or a video game console. All of these games rely heavily on visual feedback as a means of interaction with the user. Because of this, visually impaired children are excluded from this kind of entertainment.

Another problem for visually impaired children is to move with confidence in an unknown environment. It is known that blind people are able to use sound for spatial cognition [5, p1348], but it takes time and effort to train this skill and this often requires a guide.

We want to make a fun game that is suitable for visually impaired children and which is able to simulate real world environments. Then they can enjoy this way of entertainment and can train their spatial cognition as well, while playing a game in the safe environment of their home.

Problem definition

There are several challenges when making this product. The most important challenge is making the game accessible for visually impaired children. Feedback from the game must rely on other senses than sight, such as hearing and touch. In addition to this, visually impaired children must be able to operate the game. To help children operate the game, a tutorial should be available. To train the children's spatial cognition they must get a realistic audio signal as feedback from the game. This audio signal should simulate real world characteristics of audio (e.g. reflections). As we are making a game for children, there will be parents or caretakers who want to monitor the child playing the game. We as makers of the game also want an easy way to validate our product during the design process. For these two reasons there should be a visual output. And finally it is of course important that the game is still a game, i.e. it should be fun to play. These challenges can be summarised in the following problem definition:

Designing a fun audio game on the personal computer accessible for visually impaired children which simulates real world environments using three-dimensional audio.

Thesis outline

In this thesis we will further expand on our problem definition in Chapter 2, where we will specify our problem into multiple requirements. In Chapter 3 we describe the completed research in the field of audio games for the visually impaired. Which we will use in Chapter 4, the design process, were we describe how we meet the requirements. In Chapter 5 we will describe the results from testing our game. In Chapter 6 we will then discuss these results. In Chapter 7, the conclusion, we will reflect on the requirements we met and those we did not. And finally we will give our recommendations in Chapter 8. As part of the course 'Ethics and Technology', we will describe the ethical side of our product in Appendix B.

2

Problem Definition

It is important to quantify the problem definition into specific requirements. It is also important to define various subsystem of the prototype to create a balanced work load for all members of the group. In this chapter the work distribution is described and the specific requirements are listed. Only a prototype of the game will be made and therefore only the requirements for the prototype are listed.

2.1. Definition of system boundaries

The prototype will be made by a group of six students. This group is divided into three sub-groups of two students each. Thus the prototype has to be divided into three subsystems: the gameplay with the graphical user interface (GUI), the audio localisation [1] and the sound reflections [2]. We are responsible for the gameplay, the GUI and the audio output. How these subsystems relate to each other can be seen in Figure 2.1.

The gameplay is responsible for the game mechanics: simulate a player in the game, simulate an environment around the player, keep track of the score, etc. The gameplay is also responsible for the interaction with the user. The user input is processed and feedback is given in the form of the audio and the GUI.

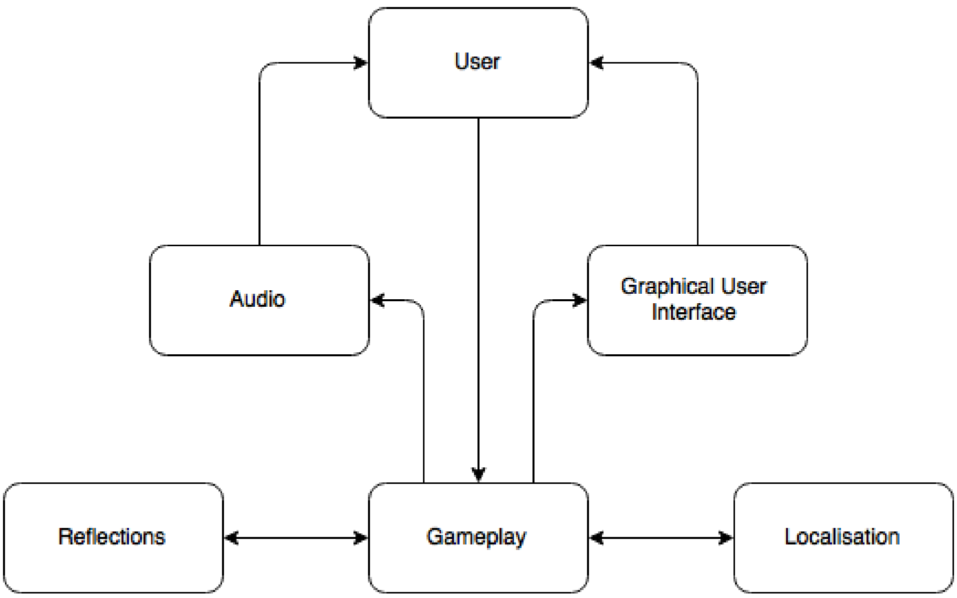

Figure 2.1: The subsystems

The audio is simulated in two steps by the other subsystems. First the room dimensions and the positions of all sound sources are send to the reflections subsystem. This subsystem returns the locations of all reflection sources to the gameplay, with appropriate filtering (wall absorption [2]). Next the location of each sound source (real sources and reflection sources) and the location of the player are send to the localisation subsystem. This subsystem returns the localised audio (the simulated sound of each source at the location of the player). The localised audio is combined in the audio subsystem and send to the user.

The gameplay subsystem is also responsible for a tutorial. In the tutorial the user is made accustomed to the game. This tutorial will use simplified audio as feedback (e.g. no reflections) to not confuse new users. Speech will have to be used to explain how the game works. Because only simplified audio is used and because speech needs to be played, the gameplay subsystem simulates the audio for the tutorial themselves.

2.2. REQUIREMENTS

Some of these requirements were made in collaboration with other members of our project group, others were given by the supervisor and the rest were made by us. Before the requirements a short explanation is given, after which the requirements are listed. The summarised problem definition is:

Designing a fun audio game on the personal computer accessible for visually impaired children which simulates real world environments using three-dimensional audio.

2.2.1. PROTOTYPE REQUIREMENTS

The game will be made for visually impaired children. The most important requirement is that they can interact with the game (Req. 1.1) and they can use methods they are accustomed to (Req. 1.2). The game must run on the personal computer (Req. 1.3), because development of programs is easy for this platform and it is widely available. Because personal computer is not a very specific definition, two extra hardware requirements are defined (Req. 1.4 and Req. 1.5). On the personal computer a programming language is required to develop the game (Req. 1.6).

The game has to simulate real world environments using three-dimensional audio (Req. 1.7). This audio can not have any stuttering, clicking or popping (Req. 1.8). The delay between user input and audio output must be small (Req. 1.9), else this would cause difficulties interacting with the game.

Caretakers or parents must be able to monitor the child playing and there must be a way to validate the game design, that is why a graphical user interface is needed (Req. 1.10). Many games have competitive elements. To achieve this, the game must have scores (Req. 1.11). Users must have a way to get accustomed to various concepts of the game (Req. 1.12).

The prototype must be developed before the end of the project (Req. 1.13) and there is no budget for the prototype (Req. 1.14).

Req. 1.1 *The game must be fully accessible for visually impaired children.*
Req. 1.2 *The controls and feedback of the game must be compatible with available methods of computer interaction for visually impaired children.*
Req. 1.3 *The game must be developed for the personal computer.*
Req. 1.4 *CPU usage must be limited to 50% on a 2.9 GHz Intel Core i7.*
Req. 1.5 *Memory usage must be limited to 500 MB.*
Req. 1.6 *The prototype must be developed in a high-level, object-orientated programming language.*
Req. 1.7 *The game must use three-dimensional audio to simulate an environment.*
Req. 1.8 *The audio output should be smooth, e.g. no stuttering.*
Req. 1.9 *Between user input and the audio output there must be a maximum delay of* 100 ms .
Req. 1.10 *The game must have a graphical user interface to show the current state of the game.*
Req. 1.11 *The game must have a scoring system to monitor the progress of the child and to add a competitive element to the game.*
Req. 1.12 *The prototype must have a tutorial to prepare new users for the game.*
Req. 1.13 *The prototype must be developed in ten weeks.*
Req. 1.14 *The prototype must be developed without additional costs.*

3

Related Research

Before we start designing our prototype, we research what literature is available about our subject. First we look at what games are available for visually impaired people and discuss the capabilities and shortcomings of a few games. Then we will describe how spatial audio works and how it can be simulated on computers. Finally, we will describe how visually impaired people interact with computers.

3.1. Games for the visually impaired

At the moment multiple audio-based games are available for visually impaired children. Many of these games are small projects of individuals and therefore have various degrees of entertainment, complexity and completeness [6]. Many of these games do not function properly or only run on outdated software. The subject of audio based games has also been researched and papers about this are available [7] [8] [9]. We will now highlight a few games that use localised sound and discuss their capabilities and shortcomings.

Where's My Rubber Ducky [10]

This game is developed in the Netherlands for mobile devices. The goal of the game is to use directional audio to locate zombies and then shoot them. The player is stationary and can only turn towards the zombies. Because of this, and because sound reflections are not implemented, this game does not simulate real world environments. Entertainment is the main focus of this game.

A Blind Legend [11]

This game is currently still in development in France and will be played on mobile devices. A demo version has been released on the personal computer. In the demo you are in a forest and you have to navigate towards your daughter who is calling for you. This game makes use of binaural sound (see Section 3.2) in the Unity game engine. Unity can simulate sound in 3 dimensional space, but sound reflections are not implemented or only consist of a simple filter [12]. Again, entertainment is the main focus of this game.

The Explorer [13]

This game is developed in the Netherlands in collaboration with *Koninklijke Visio* [3] and *TNO* (nonprofit company that focuses on applied science). The game is available on the Wii video game console. The goal of the game is to find treasures in a maze of underground passages. The focus of the game is the ability for visually impaired children to play a game together with sighted children and improve the development of their motor skills. Instead of only using simple localisation audio, the game also relies on auditory cues to the player (e.g. quicker beeping when approaching a treasure).

Access Invaders [14]

Access Invaders is a collection of different version of the classic game *Space Invaders* for disabled people. We will only look at the version made for visually impaired people. The goal of the game is to shoot aliens in a two-dimensional field, you are positioned at the bottom of the field and can only move right or left. The game uses simple localisation audio to indicate when the alien is to the right or left of the player. This way the player can position themselves and shoot the alien. The main focus of this game is accessibility and not realistic sound simulation.

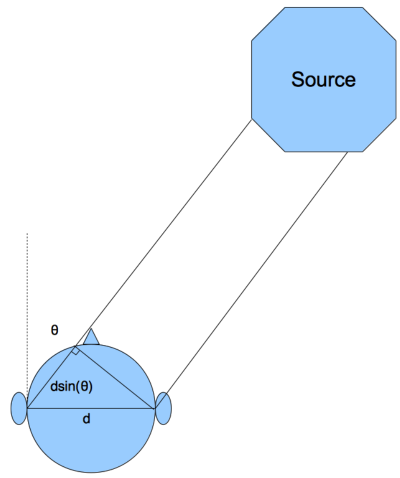

Figure 3.1: Model for the interaural time difference

CONCLUSION

It will be clear from these examples that there are multiple games available that simulate sound in one- to three-dimensional space. But none of these games accurately simulate the real world, as none of these simulate reflections accurately, while reflections do affect audio localisation [15].

3.2. SPATIAL AUDIO

We know from Section 3.1 that in the current games it is possible to simulate sound in one- to three-dimensional space. But in which way do people get a perception of the environment with only audio at their disposal? In this section we explain two aspects of audio perception: how to get a spatial impression and how to add a direction to a point sound source.

3.2.1. LOCALISATION METHODS

Temporal and intensive characteristics of acoustical stimulis provide cues about the locus of the sound, because the sound is diffracted by our head and ears. The auditory system uses different methods for localising sounds [16, 17]:

- Binaural localisation (The main tools for localisation in the horizontal plane)
- Directional filtering of the pinnae (To distinguish between front, back and different elevations, and more accuracy)
- Head movements (For more accurate localisation)

For horizontal localisation, spatial information is derived at high frequencys with interaural intensity difference and at low frequencies with interaural time difference. Vertical localisation is sensitive to manipulation of the source spectrum. For a vertical localisation pinnae cues can be used, these are spectral shape cues [17]. Because of the system boundaries only horizontal localisation will be explained in this section.

BINAURAL LOCALISATION

For binaural localisation the main tools are interaural intensity difference (IID) and interaural time difference (ITD) [16, 18]. The equation for the time difference due to the angle of the sound is:

$$\Delta t = \frac{d \cdot \sin(\theta)}{c} \tag{3.1}$$

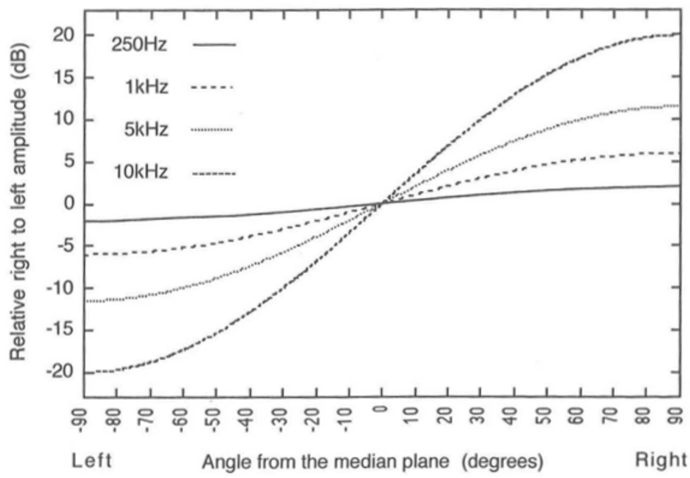

Figure 3.2: IID as a function of angle and frequency with data from Gulick 1971 [16, p. 102]

In this formula Δt is the time difference (in s), d the distance between the ears (in m), θ the angle of arrival (in rad) and c the speed of sound (in ms^{-1}) (see Figure 3.1). For sine waves, the ITD can also be described as a phase delay. In this formula the sound travels through the head, so this is not the most accurate formula for the time delay. When the phase shift is greater than π radians there is a problem with the direction because there are two possible angles. The maximum frequency where ITD can be used before this problem arises, is given by Equation 3.2 [16], where r is half the distance between the ears (in m) and θ the angle of arrival (in rad).

$$f_{max}(\theta) = \frac{1}{2r \cdot (\theta + \sin(\theta))} \tag{3.2}$$

The other difference is interaural intensity difference. Because of the shading effect of the head, the intensity of the ear furthest away from the source has more reduction of the sound than the other ear [16]. At low frequencies the wavelength is long and the sound 'bends' very well around the head. At high frequencies there occurs little diffraction [19]. This is hard to calculate, but there are experiments of the reduction in dB at different frequencies [16], see Figure 3.2. The minimum frequency where IID is useful for localisation correspond to when the head is about two third of a wavelength in size, because otherwise the head will tend to scatter the energy in all directions and there is very little diffraction. This can be described by Equation 3.3 [16], where c is the speed of sound in (m/s) and d is the diameter of the head (in m). The cross-over of IID and ITD starts around f_{max} of ITD and f_{min} of IID to about four times this frequency [16].

$$f_{min(\theta=\pi/2)} = \frac{1}{3} \cdot \frac{c}{d} \tag{3.3}$$

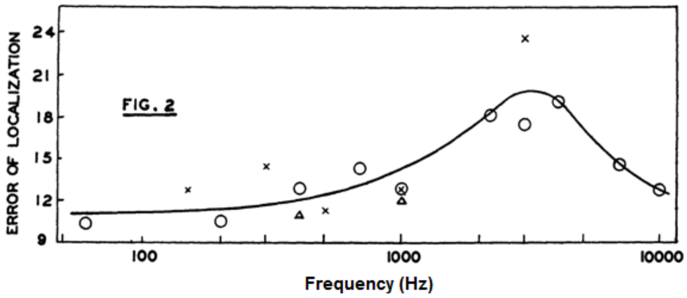

Figure 3.3: Average errors in degrees with different frequencies [20]

Research on localising sources of sound have shown how we localise sound and in which frequencies localisation works best. In Figure 3.3 we see that with low frequencies and high frequencies the error rate is lowest. Around the 3 kHz the error in degrees is the highest, because of the absence of interaural phase difference and interaural time difference [20, p. 300-303]. Also according to Stevens and Newman the error of localisation is smallest for tones located near the median plane and increases when going to the side [20, p. 306].

3.3. COMPUTER USE FOR VISUALLY IMPAIRED PEOPLE

To develop a game that is aimed at visually impaired children we have to know how the visually impaired use the computer. According to the research by prof. Reeta Singh, blind people do not use a mouse, because the mouse is based on sight, but they primarily use the keyboard. Blind people can learn how to use any program, as long as all functions can be accessed with a keyboard. The visually impaired and blind people have a limited range of tasks they can do on a computer. The main problems for blind people are tasks that are mainly graphical or involve interaction with moving images [21].

There are many solutions for easy computer access and web access for the visually impaired. These solutions can be separated in auditory output and tactile output [22]. For auditory interfaces the first step is to convey information about the individual objects [23], with for example auditory icons [24, p. 1025] (everyday sounds mapped to computer events). For tactile output there are for example braille displays that give feedback to the user [22].

4

Design Process

In this chapter the design process is described. The information from Chapter 3 is used to complete the requirements which were set in Chapter 2. The design process can be divided into several sections:

- The type of game we will be making.
- The programming language in which we develop the game.
- Gameplay; all the game mechanics.
- The game controls for the user.
- The realistic simulation of audio.
- The various audio clips to give information to the user.
- Playing the audio in real time on a computer.
- Visual display of the game.

4.1. Type of game

The first step in the design process was the decision what type of game we would be making. In our assignment it was specified that it should be an audio based snake game, but we were given the freedom to choose another game, as long as it utilised localised audio. In this section we will briefly describe some of the options we considered, their pros and cons and then explain our choice for the type of game.

4.1.1. Options

Snake

The goal of this game is to manoeuvre a snake in an enclosed two-dimensional grid, which grows in length when objects are 'eaten'. As the game consists of a discrete two-dimensional grid, implementation is easier compared to more continuous game environments. The object that needs to be eaten would act as a sound source, so you would need to use localised audio to find it. The disadvantage of the two-dimensional grid is that rotation is limited to 90 degrees increments, so you can not move naturally through the game environment and the direction of the audio makes sudden changes. Another disadvantage of implementing this game with audio is that you have to indicate where the tail of the snake is. When the snake grows it becomes very complex to give this information with only audio feedback.

Pong

Pong is a simple two-dimensional table tennis simulation, the goal is to make the opponent miss the ball which equals scoring a point. As this is one of the first video games ever released, implementation would not be very difficult. The ball would be an audio source and the user would hear this sound from the position of the paddle. An advantage of this game is that you would be able to play against a sighted person.

DUNGEON CRAWLER

In this game the player moves through a dungeon (maze) environment to find treasures while fighting monsters. We could implement this in a three-dimensional game environment, which is hard to implement, but very immersing for the user. The treasures and monsters would both be sources of sound, which you would need to respectively find and avoid. As this is a 'realistic' game environment, physical attributes of sound (like reflections) can be implemented.

OBSTACLE COURSE

The goal of this game is to avoid obstacles while the player moves on a track. Instead of moving towards sound sources you would have to avoid them, as obstacles would emit sound. The advantage of this game is that it is easy to implement, as the game is not very complex. But because of this simplicity the game environment is not very real world like and movements are limited to one-dimension.

4.1.2. OUR CHOICE

We choose the Dungeon Crawler type of game, mainly because it has the least disadvantages and seemed like a good challenge to make. The biggest disadvantage is the high complexity, this is why we simplified this option. We choose to focus on the basics of such a game: a single room with a player and a treasure, the goal is to move the player towards the treasure. An advantage of this type of game is that it has an overlap with our original assignment, as it can be seen as a type of snake with a non-discrete game environment.

4.2. PROGRAMMING LANGUAGE

According to Req. 1.6 a high-level, object-orientated programming language must be used. This language should also be free, as the prototype has to be developed without additional costs (Req. 1.14). We chose to use Python [25], an open-source programming language. This decision was made with the entire project group. The reasons of the other subgroups for this decision can be found in [1] and [2]. For us this meant we could use a dedicated package for game development: *PyGame* [26]. This simplified development compared to for example MATLAB, which does not have such a package.

4.3. GAMEPLAY

We have chosen to make a Dungeon Crawler game, which results in a few requirements for the gameplay. We need a player that is controlled by the user, a treasure to find, an environment in which these two are located and a way to keep track of the score. We also need different levels for a progression of the difficulty in the game. In this section we will cover the game mechanics. When we have the building blocks for our game, we still have to decide how to use these blocks to make a game. After the game mechanics we describe the level design in our game. We first design a few tutorial levels to make the user accustomed to the game, after which we describe a few levels we implemented. Finally, we describe our design of a menu to select a tutorial or level.

4.3.1. GAME MECHANICS

THE PLAYER

The player is directly controlled by the user. The position of the player is also the position from which the user hears the sounds from the game. Because of this, the position of the player is defined as the position of the head of the player, this is done in three dimensions, so players can have different heights. The player also has a rotation, which is the direction the player is looking. For simplicity, the direction of the head and the body of the player are always the same, which means you will move forward in the direction you are facing. The player moves at walking speed in the game.

THE TREASURE OR SOURCE

The treasure is the target in our game. In the prototype the treasure only acts as a sound source, so it will be called the source from now on. This source emits a sound to help the user localise it. The position of the source is also defined in three dimensions, so it can for example be fitted to the ceiling. The objective is to move the player into close proximity of the source.

THE GAME ENVIRONMENT

The player and the source are located in the game environment. As we are making a Dungeon Crawler, the environment consists of enclosed rooms of different shapes and sizes. This enables us to define the environment by the walls that surround it. The player and the source are always placed inside the walls and are not able to move through the walls.

THE SCORE

There are multiple methods to implement a scoring system. The goal of the game is to find the source, so the simplest method is to make the user complete the level by finding the source once. But as this would make levels very short in our simplified Dungeon Crawler we looked to more complex methods for a scoring system.

In our first implementation the user has to find the source, at which point a 'hit' is added, the old source is removed and a new source is placed in the room. After the user does this for a pre-specified amount of times, the level is finished. During the level a timer is run, this time is the score of the level.

A second implementation is a speed challenge. A timer starts at a pre-specified time and counts down to zero. When zero is reached the level is over. But every time a source is reached, a pre-set amount of time is added to the timer. When the time runs out, the amount of times the source is found equals the score.

We want a method to compare scores with scores of other people or your own highest score. We therefore implement high scores. At the end of the level the user can enter his name and the score is then saved. Each level has its own high scores, but only the ten highest scores are saved.

4.3.2. TUTORIALS

We implemented two tutorials for our game. The objective of these tutorials is to make the user accustomed to some of the game mechanics, i.e. the audio localisation and player movement. Every tutorial focuses on one of these game mechanics.

TUTORIAL 1 - LOCALISED AUDIO

In the first tutorial the user will be made accustomed to the localised audio. In this tutorial the player is stationary and cannot be moved. The source is positioned in five discrete locations at a constant distance from the player, see Figure 4.1. We decided to bias the diagonal positions to the center (30° from center, instead of 45°), this was because the difference (of the localised audio) between center and slightly off center is larger than the difference between to the right and slightly to the front right. This effect is caused by the intensity difference for various angles (see subsection 3.2.1). First the source is moved through these positions in an anticlockwise direction. Then the source is positioned (randomly) on one of these locations and the player has to specify the direction from which the sound came. When the player has done this correctly five times, the tutorial is completed.

TUTORIAL 2 - PLAYER MOVEMENT

In the second tutorial we want to make the user accustomed to the way the player moves in the game. The player position is initialized with a sound source to the right of the player. The user is instructed to turn the player towards the source with the appropriate controls. When the player is facing the source, the user is instructed to move forward towards the source and adjust the rotation when necessary. When the source is reached, the tutorial is over.

4.3.3. LEVELS

We were limited in the diversity of levels by the fact that reflections can only properly be modelled for concave rooms (all angles are smaller than π) [2]. We designed our levels in such a way that as many features as possible would be showcased.

LEVEL 1

In the first level the player starts in the middle of a square room (8x8 meters) with a source randomly placed. As a scoring system we use the first implementation, the user has to find the source five times to complete the level.

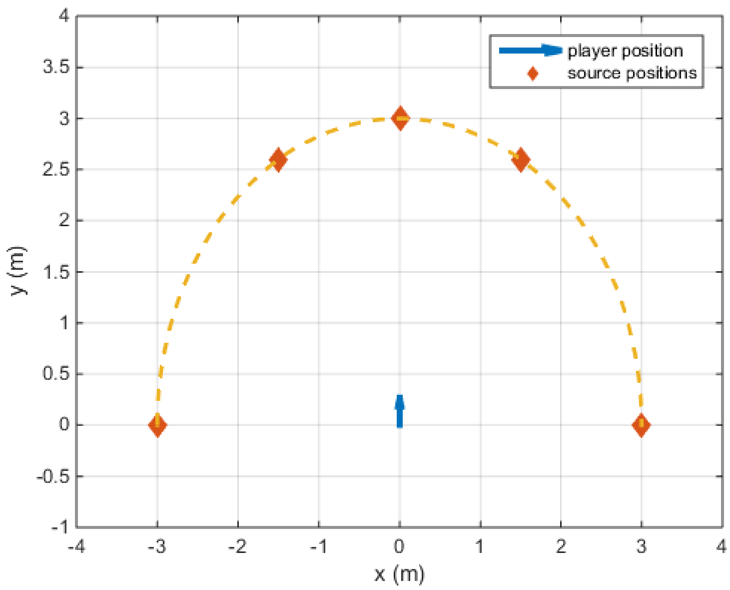

Figure 4.1: Five possible positions for the source in tutorial 1

LEVEL 2

In the second level the player start in the middle of a octagonal room (with sides of 4 meters) with a source randomly placed. This time we use the second implementation of a scoring system (speed challenge). The user starts with 20 seconds and every time a source is located 3 seconds is added to the time. When the time reaches zero the level is over.

4.3.4. MENU

We implemented a menu so the user is able to choose a level to play. When the game starts this menu opens with the title selected. The user can then switch to a level (or tutorial) and when he confirms his selection this level starts. Once the level is completed, after the high scores, the menu opens again.

4.4. GAME CONTROL

The user has to be able to interact with the game, in this section we will cover our choices on how to do this with visually impaired people. We were limited in our choice of interaction by Req. 1.2 and thus the way visually impaired people interact with computers (Section 3.3). The mouse is difficult to use for visually impaired people and we did not want specialised hardware being a requirement for the game. With this in mind we choose to exclusively use the keyboard as input device.

The question that now remains is how to use the keyboard to control the flow of our game? We choose to further limit the input, to keys that are easy to find (like arrow keys, space bar, escape, etc.). This way our game is accessible even for children that have not completely learned how to use the keyboard.

When the game starts the menu is opened in which the user must make a selection, this is done by using the up and down arrow keys to select something and then pressing the return key to confirm the selection. In a level the user must control the player, the arrow keys are used for this. The right and left arrow keys rotate the player while the forward and backward arrow keys move the player forward and backward. Multiple keys can be pressed at the same time, e.g. forward key and right key to turn the player to the right while moving forward. In tutorial 1, the player can not move, but a direction must be specified by the user. This is done with the arrow keys (for diagonal position this means pressing forward arrow key and left or right at the same time). The last control we implemented for visually impaired users is the escape key, with this key a level can be stopped or the game can be closed (only from the menu).

4.5. AUDIO SIMULATION

For the audio simulation of the game we use localisation methods and reflections to give the player the experience that he is in a room with a source that makes a sound. In tutorial 1 the goal is to learn to localise, therefore we add only the most important methods for localisation. In tutorial 2 and in the levels we use the localisation subsystem [1] and the reflections subsystem [2]. This way the audio is more realistically simulated when the player can move.

4.5.1. TUTORIAL 1

In tutorial 1 the basics of audio localisation are used. The player is stationary and the sources are positioned in front of the player. The method that we use is binaural localisation. For this method interaural time difference (ITD) and interaural intensity difference (IID) are implemented. Equation 3.1 is used for the ITD and for the IID the intensity difference around the frequency of 5 kHz is used, see Figure 3.2. We have chosen 5 kHz because this is a high-frequency which has little diffraction. Figure 3.2 is made with experimental data, this means an equation has to be found that closely resembles the data. The arctan resembles the data and with two constants (experimentally determined) we have an equation for the IID (see Equation 4.1). θ is the angle from the median plane in radians. As can be seen in Figure 4.2, this equation closely resembles Figure 3.2 for 5 kHz.

$$y = 10 \cdot \arctan(\frac{4\theta}{\pi}) \tag{4.1}$$

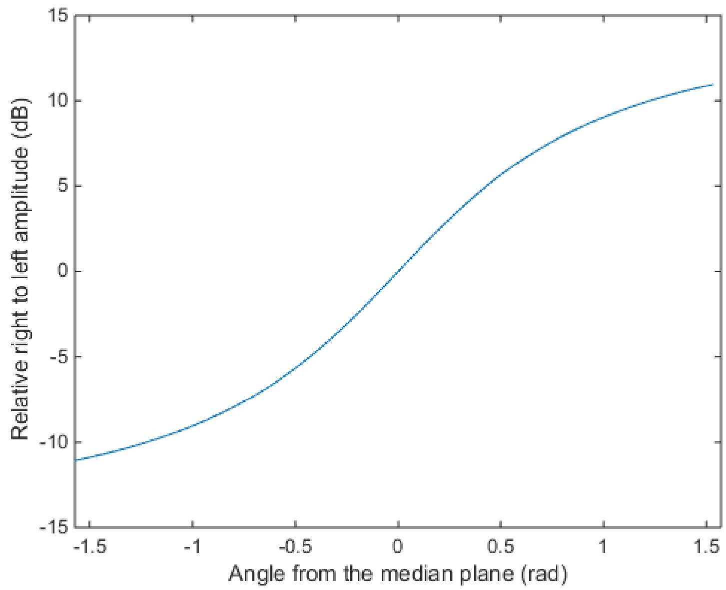

Figure 4.2: IID as a function of angle at 5 kHz

In tutorial 1 we have chosen not to implement the reflections, because of the goal of learning the principle of auditory localisation. Reflection is adding an impression of the space where the player is and it gives only information of localisation of the source when it is in the same direction of the source Section 3.2.

4.5.2. TUTORIAL 2 AND THE LEVELS

In tutorial 2 and the levels the player is in a environment which consists of enclosed rooms of different shapes and sizes, where the player can walk freely. For the localisation we use a module described in [1]. This module consists of: binaural localisation, filtering of the pinnae and head movements. Binaural localisation is what we also use in the tutorial. For better localisation filtering of the pinnae and is added to the game. For the reflections we use a module as described in [2]. In this module we can get all reflections for sources in a room with angles equal or smaller than π. For our game we only use the first order reflections. This because we have limited time to calculate all the reflections Req. 1.9.

4.6. AUDIO CLIPS

For the game there are three things important to know as a user: location of the source, location of the walls of the dungeon and the location of the player. It is not necessary to know exactly where the player is, but the player is walking in a room and therefore with good audio simulation we want to give feedback when the player for example walks to the wall. Visually impaired children also have to navigate in the menu and know how to play the game. This can be done with speech for every text in the game and instructions before every level. For the non-speech audio clips we make use of an audio database released under the Creative Commons license [27].

4.6.1. GAME AUDIO

The audio clips of the game can be separated into four categories:

- Source
- Dungeon
- Player
- Miscellaneous

SOURCE

The source has to be a sound that contains low and high frequencies, so the localisation methods IID and ITD can both be used to localise the source. Also the sound has to be continuous so the player can always localise the source.

For the source we considered different clips, therefore we made spectrograms of the clips. The first clip is an audio clip with violin, bass and drums to represent the low and high frequencies. The second clip is an audio clip with cello, choir and piano. In Figure 4.3 we see that for the first clip, the power spectral density for the different frequencies is almost the same for 200-6000 Hz. In the second clip the low frequencies are mainly present. In the second clip the power spectral density of the frequencies 4-10 kHz are much lower in comparison to the lower frequencies. The first sound clip is therefore better, because IID and ITD can both be used.

DUNGEON

Of course a dungeon is not completely quiet, there are background sounds. But we choose not to implement these in the prototype, because these would only act as interference in the localisation. This is something that would be useful in the final product when more difficult levels are needed.

PLAYER

For the user it is important to know what the location of the player is relative to the source but also to the room. The location relative to the source can be obtained with the localisation methods between source and player. For the location in the room we can use the reflections. Therefore we add footsteps, like in real life, to the player when he walks.

MISCELLANEOUS

Normally the progress in a game is displayed on the screen. In the way we designed the game, the feedback can only use audio for the visually impaired. The audio clips indicating progress in the level can be very short because it is not always needed and with recognisable clips the player can easily understand how far he is in the game, for example with auditory icons. Therefore we added for every successful search to the source a success clip. To know that the game is ended, there is also a short audio clip added.

4.6.2. SPEECH

The game has a menu where the player can navigate between the tutorial and the levels. We know from Section 3.3 that visually impaired also use audio for navigating on a computer. For auditory navigation we have to give every individual object an audio clip. We choose to do this with speech clips which tell the user what menu item is selected.

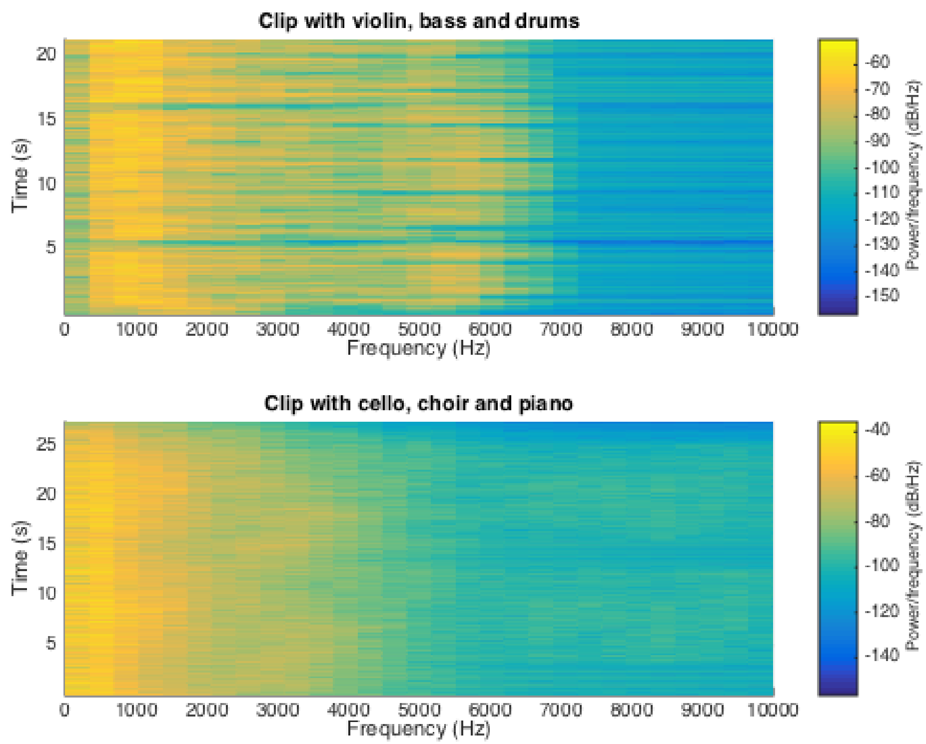

Figure 4.3: Spectrogram of two soundclips

When accessing the different tutorials and levels there must be an explanation what the goal of that tutorial or level is and how to play it. We were not able to implement this in the game, due to the time limit of the project (Req. 1.13)

4.7. AUDIO PLAYER

The audio player is the only output of the game for visually impaired children. Because of that, we want to design an audio player that gives continuous sound and where the localised audio is always up to date. To have the audio player always up to date there is a buffer needed to add new sound vectors for the audioplayer. When moving the player there is a 'discrete' version of the Doppler effect, which needs to be smoothed. The solution to this problem can be found in [1].

4.7.1. BUFFER

The sound vector to the output of the audio player is changing constantly because the player location changes. The processing of the sound has to go via a buffer to make the audio processing continuous. The delay can not be larger than 100 ms according to Req. 1.9. This means we have to choose a maximum frame size with a sample rate for the audio player. In Equation 4.2 the formula to calculate the delay is shown. Δt is the delay in seconds, the frame size is the number of samples in one frame and the sample rate is the number of samples per second. The samples go to a buffer where the audio player sends them to the output a frame size later.

$$\Delta t = \frac{\text{frame size}}{\text{sample rate}} \tag{4.2}$$

Most music files have a sample rate of 44.100 kHz, with a maximum delay of 100 ms the frame size can at most be 4096 samples. We use a frame size of 1024 in our prototype to minimise the delay. But when a computer has difficulty running the game, this frame size could be increased.

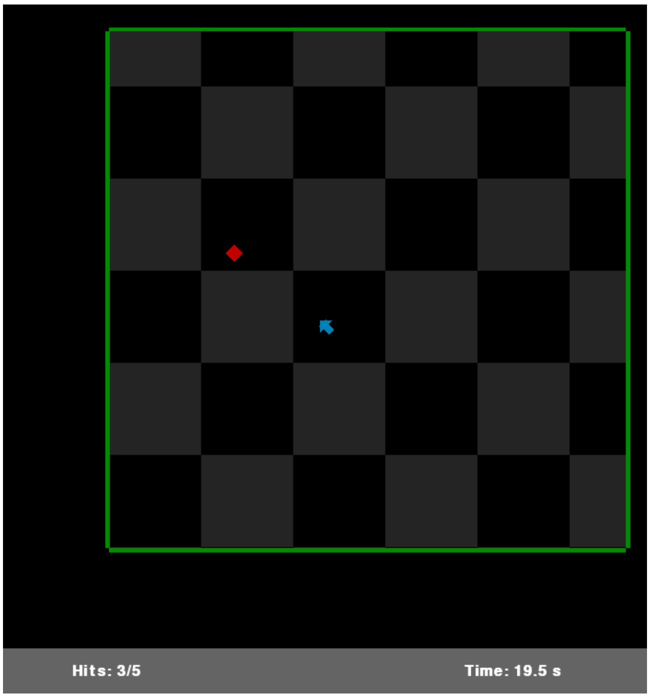

Figure 4.4: Graphical User Interface while playing a level

Figure 4.5: Player Image

Figure 4.6: Source Image

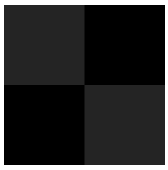

Figure 4.7: Floor texture

4.8. GRAPHICAL USER INTERFACE

One of the requirements of the prototype (Req. 1.10) is a graphical user interface (GUI). In this section we will describe the GUI we designed. We will regularly reference images of the GUI in Appendix A.

THE MAIN MENU

The graphical representation of the menu can be seen in Figure A.1. All possible selections are visible and the currently selected item has a grey square beneath it to indicate this.

THE LEVELS

An impression of the GUI while playing a level can be seen in Figure 4.4. The window is divided in two sections, the main part shows the game environment with all the elements (player, source and walls) and a bar at the bottom shows information like the current score. The main part can be made invisible (with the *'b'* key) when sighted people are playing the game to prevent these people using their sight as an advantage.

For the levels we make use of various images to represent the game elements. For the player we use a blue arrow (Figure 4.5), this way we can indicate position and rotation of the player. For the source we use a simple red diamond, see Figure 4.6. The graphical representation of the game environment is always centered on the player, this way rooms can be bigger than the screen. In very big rooms it is not clear if the player is moving though, so we added a checkered floor texture (see Figure 4.7) to make this clear. In Figure A.2 to Figure A.5 the graphical representation of the various tutorials and levels can be seen.

HIGH SCORES

After a level is completed the player can enter his name for the high scores, this is also made visible in the GUI (Figure A.6). Next the current high scores are shown on screen (Figure A.7), after which the user returns to the menu.

5
Results

The game must be tested to validate the design. In this chapter we will describe the methods we used to test our game. We tested our game with visually impaired children at *Koninklijke Visio* [3] and students at Delft University of Technology. The performance of the game on the specified hardware (Section 2.2) was also tested.

5.1. Testing with users

In this section the tests with users will be described. A test plan is made and this is used to get comparable results from the test subjects.

Test plan

Before we test if the game works, we want to define a standard method of testing. This way we can compare the various results with each other. In this section we will define a test plan which will be used when the subjects play the game.

Our test subjects are given a headset and a keyboard, the graphical user interface (GUI) is not available for them. They are then instructed to play tutorial 1 and 2. When they have completed these tutorials they are instructed to play level 1 twice. As speech is not fully implemented, we read the appropriate instructions to them. To read the instructions at the appropriate time, we monitor the GUI. We keep track of things that go well or wrong and we keep track of the score the users set.

Results

First we describe the results of testing with visually impaired children. Then we describe the results of testing with students. The students were divided in two groups: members and non-members of our project group.

Visually impaired children

We were able to test the game with visually impaired children at the *Koninklijke Visio* [3]. Unfortunately only two children were available. They are both around ten years of age, the difference in age between them was approximately three years. To prevent the children getting frustrated with the game, we decided to give additional feedback if they would get stuck. This additional feedback consisted of short instructions how to reach the source (e.g. “turn a bit to the right”, “maybe move a little forward”).

Tutorial 1 For both test subjects it was difficult to recognise the sources with an angle of 30° degrees to the right or left. In addition to this, it was difficult to indicate this direction with the keyboard, as two keys had to be pressed at the same time.

Tutorial 2 They both recognise the audio changing when they change directions. However, they both had difficulty walking towards the source. Especially when they were far from the source, it was difficult to get back to the source.

Level 1 To play the game they needed additional feedback. Test subject A (younger child) had difficulties with the controls. For example while moving towards the source, A was unable to turn while walking or stop walking when close to the source. Instead test subject A often walked until a wall was reached and only then the direction would be corrected. Test subject B (older child) had problems in the beginning with recognising walls and needed additional feedback to correct this. However, the second time B played the additional feedback was not needed. Test subject A did not recognise intensity differences when walking towards or away from the source. Test subject B was able to hear this difference. In Table 5.1 the results of Level 1 are shown. Only test 2 of test subject B is done without additional feedback.

Test subject	Test 1	Test 2
A	209,6	222.7
B	289.1	197.6

Table 5.1: Highscores of the visually impaired children for Level 1 (in seconds)

STUDENTS

We could only test with two visually impaired children, therefore we also tested with students at the TU Delft to know if the game can be played without our feedback and only audio as feedback. For this test group we used the same test plan as for the visually impaired children, except no additional feedback is given. We divided the students into two groups: students that were not members of our project group and those that were. Members of our project group had played the game before, which meant they already had experience.

Tutorial 1 No problems were encountered in this tutorial. They knew what they had to do and recognised the location of the sources.

Tutorial 2 For this tutorial we can divide all the students in two groups. The first group mastered the controls quickly. For the other group it was not very clear how to control the player in the beginning. This meant it took more time to finish the tutorial.

Level 1 In Table 5.2 and Table 5.3 the results of Level 1 can be seen. The test subjects in Table 5.2 are students who played the game for the first time. The test subject in Table 5.3 were members of our project group, they had played the game before. Test subject G in Table 5.2 had trouble understanding the game during test 1. Between test 1 and 2 we therefore gave G a short recap of the tutorial instructions.

Test subject	Test 1	Test 2
A	67.4	35.9
B	136.7	113.0
C	32.6	33.6
D	77.2	42.7
E	86.9	64.3
F	78.8	96.4
G	203.5	35.7
H	37.3	19.8

Table 5.2: Highscores of students (first time users) for Level 1 (in seconds)

Test subject	Test 1	Test 2
A	55.2	16.6
B	48.0	30.5
C	66.0	27.7
D	26.9	17.5
E	77.9	41.5
F	47.4	42.0

Table 5.3: Highscores of students (experienced users) for Level 1 (in seconds)

5.2. HARDWARE PERFORMANCE

The game needs to be tested on hardware performance. It has to be fully playable with the requirements given for the hardware. Therefore a test plan is made for the different requirements and results are shown in this section.

TEST PLAN

The tests of the hardware performance can be separated in three different tests. The first test is for CPU usage. In both levels the CPU usage will be measured every ten seconds for two minutes. The second test is for memory usage. The game and level 1 will be started five times and each time the memory usage will be measured. The last test is for the delay of the sound. We test for what minimum audio frame size the audio is still played smoothly. These tests have to be done with a 2.9 GHz Intel Core i7 to meet the requirements set in Section 2.2.

RESULTS

The first two tests are performed using a program of the operating system that measures all processes running on the PC. The result of the first test is shown in Table 5.4 and of the second test in Table 5.5. We noticed a memory leak when restarting a level without closing the game. Every time a level is started, 50 MB is added to the memory usage. For the third test the following were obtained. In level 1 the minimum frame size for which the audio is smooth, is 512 samples. This equals a delay of 11.6 ms (Equation 4.2). For level 2 the minimum frame size is 1024 samples, this equals a delay of 23.2 ms.

Seconds	**Level 1**	**Level 2**	**Seconds**	**Level 1**	**Level 2**
0	34.9	49.4	70	36.4	47.6
10	36.4	48.1	80	36.4	47.9
20	36.3	48.8	90	35.9	48.7
30	35.3	49.0	100	36.3	48.3
40	36.5	48.6	110	37.5	48.3
50	36.6	49.1	120	36.8	48.3
60	36.8	48.4			

Table 5.4: Percentage of CPU usage in the levels

Test	**Level 1**	**Level 2**
1	197.7	197.1
2	199.8	196.7
3	197.2	197.5
4	195.8	195.1
5	196.3	197.4

Table 5.5: Memory usage of the game in MegaByte

6
Discussion

In this chapter we will discuss the results of Chapter 5. We will interpret the statistical findings and draw conclusions from the observations we made.

6.1. Testing with users

Almost all users were able to complete level 1 on their own, this means the game interaction method (keyboard and audio) works. Only test subject A of the visually impaired children was unable to complete level 1 without additional feedback. This can be attributed to the young age, as young sighted children also have difficulty controlling a game.

As explained in Chapter 4 the source is randomly placed in level 1. This means no conclusions can be made about small differences (approximately ten seconds) in the results for level 1. Some users were lucky, as two consecutive sources were be placed very close to each other.

Visually impaired children

Unfortunately we could only test with two visually impaired children. Therefore conclusions we draw from our results require additional research to confirm them. Both children played the tutorials well. However, in level 1 controlling the player was difficult. The controls are too difficult for young children. They both were unable to use two keys at the same time (e.g. move forward and turn at the same time). Tutorial 2 might not be clear enough to explain the controls to new users. It is also possible more practice is needed for new users. Test subject B was able to play level 1 the second time without additional feedback. The difference in age probably plays a large role in the differences between the two test subject.

Students

For all students it was clear how the tutorials had to be played. A part of the test subjects had some difficulty understanding the controls in tutorial 2. The students who regularly play video games had no problems with the controls. For almost all test subjects there is a large difference between the first and second test. Some users did not fully grasp the game after the two tutorials, which explains part of this difference. But the members of our group (who understand the game) also have a difference between the first and second test. This indicates users might have to get used to the game every time they start playing. It could also indicate even experienced users are able to improve their performance in the game.

6.2. Hardware performance

From the first test for the hardware performance it can concluded that there is a difference in CPU usage between level 1 and level 2. This has probably to do with the design of the room. Level 2 contains eight walls and therefore more calculations have to be done (more reflections in the room).

The memory usage of the game is the same in the different levels. However, we noticed that the memory usage increases when a new level is started (without closing the game). This indicates a memory leak in the code somewhere. We think the audio is not properly closed when a level ends. This results in more data being added to the game every time a level is started. However, more testing is required to confirm this.

From the last hardware performance test it is clear that the delay can be very small. But again it depends on the complexity of the level. The minimum delay in level 2 was twice the size of the delay in level 1. When more complex levels are made, the minimum delay is expected to become larger. However, according to subsection 4.7.1, the frame size can be four times the size of the current frame size in level 2.

7

Conclusion

In this chapter the conclusions of the project will be presented. We will reflect on the requirements we set in Chapter 2. We will describe the requirements we met with our design and those we did not. We will also briefly summarise the conclusions of our results.

The game was not fully accessible for visually impaired children (Req. 1.1). The speech was not completely implemented, so they can not play the game independent of sighted people. A keyboard is used as controls and headset as feedback, therefor we met Req. 1.2. The game is developed for the personal computer (Req. 1.3) and the hardware limits have not been broken (Req. 1.4 and Req. 1.5). The prototype has been developed in Python, a high-level, object-orientated programming language (Req. 1.6).

We were successful in utilising three-dimensional audio (Req. 1.7) as feedback to the user. The audio is smooth and the delay is within the limit on the specified hardware (Req. 1.8 and Req. 1.9). The game has a working graphical user interface (Req. 1.10), scoring system (Req. 1.11) and tutorial (Req. 1.12). Unfortunately the tutorial was not adequate to prepare all new users for the game. The prototype has been finished in less then ten weeks (Req. 1.13) and without additional costs (Req. 1.14).

Overal the project was a success and a working prototype has been made. It is possible to simulate an environment using only localised audio. And using this method of feedback, it was possible for test subjects to play the game. Because of these successes, we hope our work will be used to make this world a better place for visually impaired children.

8

Recommendations

Now that the prototype is complete, we would like to make some recommendations based on what we learned during the project. In this chapter these recommendations will be discussed.

Specific target group

We defined visually impaired children as our target group. However, it became apparent this target group was not specific enough. We noticed a large difference in performance between test subjects of different age. An older child is more capable to understand complex methods of interaction with a game. While the interaction has to be very simple for younger children. We recommend to define a specific age range as a target group. This way the interaction can be made compatible with the abilities of children in that age range.

Different game control

Most test subjects had difficulty learning the controls of the game. Only people who regularly play games had little trouble. As children are often not experienced gamers (especially visually impaired children), we recommend to research the possibilities of dedicated hardware. Other methods of controlling the player may feel more natural for the user.

Speech

Currently we have not implemented speech completely in the game. This means the game is not fully accessible for visually impaired children. It would be a big improvement to implement speech in all aspects of the game. For the speech it is also very important to define a specific age range as a target group. The amount of feedback needed depends greatly on the age of the user.

Optimisation

During development we have not given optimisation much attention. We met the hardware requirements we set, but we noticed more complex levels would require a lot more of the hardware. If development of this game is continued we recommend to optimise the code for performance.

If our code is further developed it will also be important to find the reason of the memory leak. Every time a level is started 50 MB is added to the memory usage. We expect the audio is not properly closed when a level is finished, but more research has to be done to confirm this.

A

GRAPHICAL USER INTERFACE IMAGES

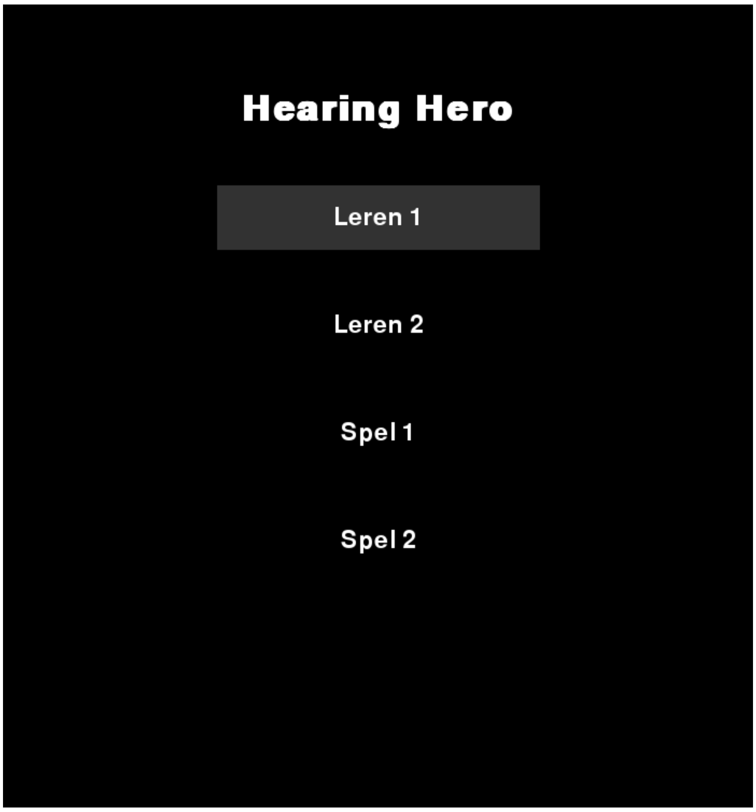

Figure A.1: The menu of the game

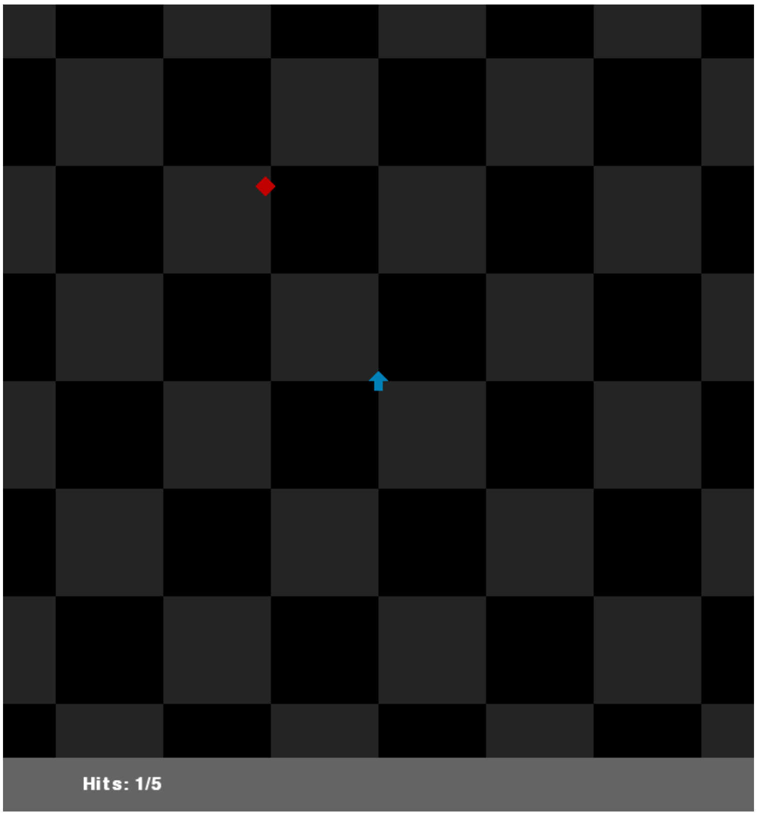

Figure A.2: Tutorial 1

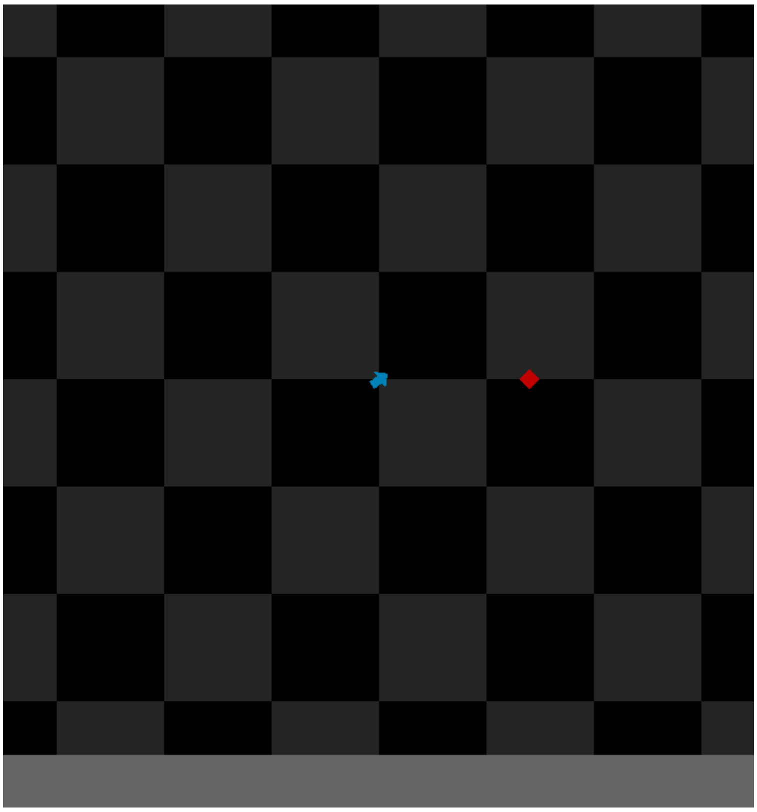

Figure A.3: Tutorial 2

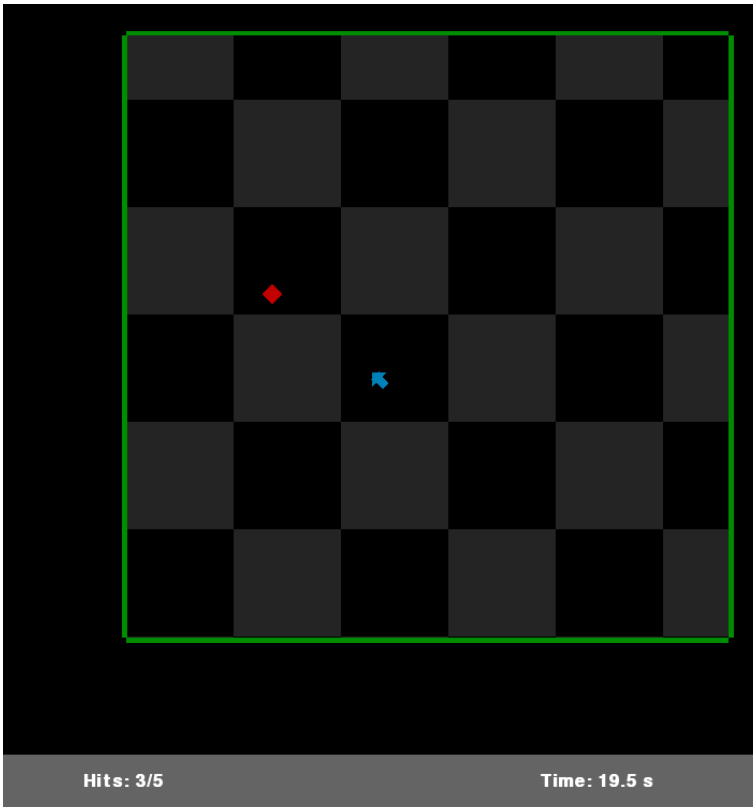

Figure A.4: Level 1

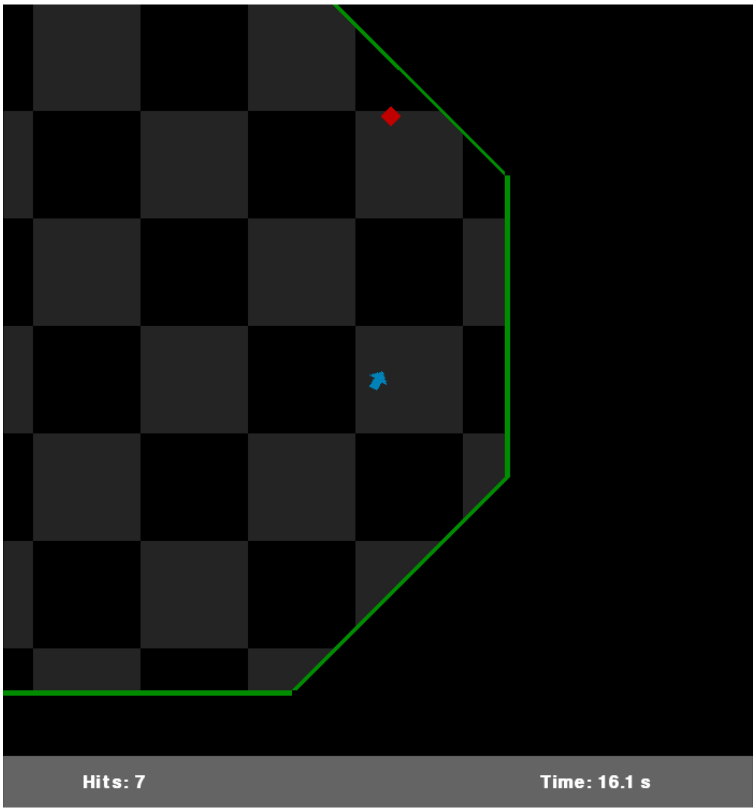

Figure A.5: Level 2

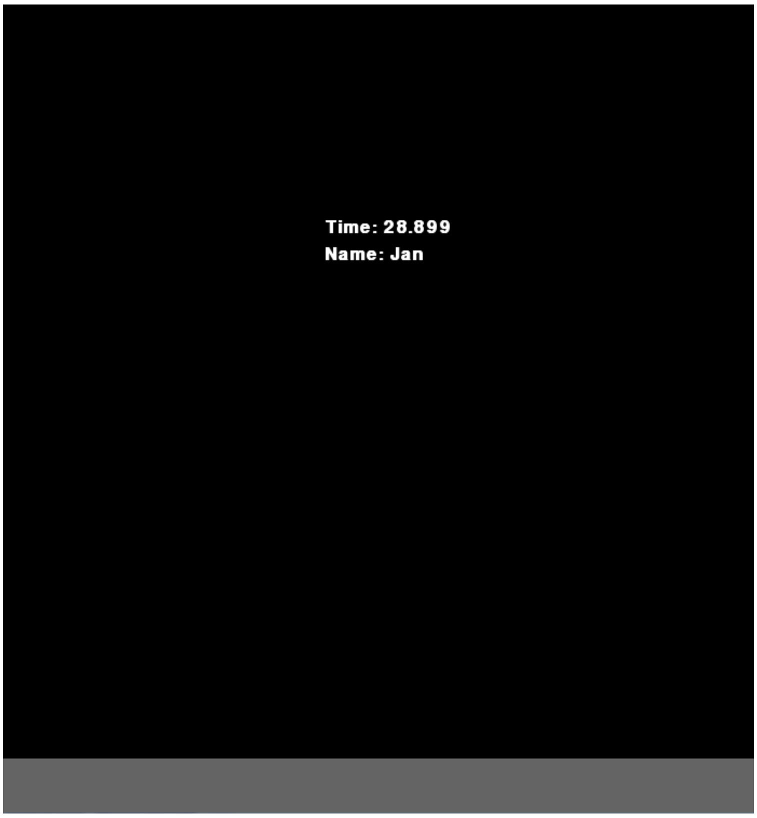

Figure A.6: Enter name for high scores

Figure A.7: List of ten highest scores

B

ETHICS

As part of the *'Bachelor Afstudeer Project' (BAP)* we did a course on *'Ethics and Technology'*. This course consists of several assignments, one of these is writing an ethical evaluation of the product we have made a prototype for. In this section we give this ethical evaluation in two parts. First we evaluate our interaction with the world (customers, people effected by our product directly or indirectly) and second we evaluate the internal structure of our company.

INTERACTION WITH THE WORLD

Most products have a social impact when they are released. Our product will have a limited social impact, as the target group of our product is relatively small. The biggest social impact will be the fact that visually impaired children can train their auditory perception on their own, instead of with a guide. The children will also spend more time on a computer then before.

The (ethical) value of this product is the fact that visually impaired children will be more equal to sighted children, as they are able to play computer games too. But this also has a negative side, as there is the possibility of a gaming addiction. But for solutions we only have to look at gaming addiction of visually able children and the methods already available to solve this.

We have to consider both our stakeholders (we will be part of the stakeholders) and the customers when pricing our product. Stakeholders will want to make the largest possible profit. But when prices rise, some parents might not be able to afford our product and their children would still be excluded from playing video games. We think it is important all visually impaired children are able to play the game, but we will not find stakeholders when there is hardly any profit. We will try to find the middle ground here, but when that is not possible we will choose the side of the stakeholders. Without stakeholders we can not make a game for any one. We do not break any rules with this choice. Thus this choice is allowed according to the deontological ethics.

We take the safety risks of our product very serious. As our game is audio based there is the risk of damaging the hearing of the user. We can not foresee people using amplified headphones, but we can make sure the volume levels are in acceptable levels during normal interaction with the game. We think we are bound to do this by the virtue ethics, as injuring other human beings is not a virtue.

As our product is purely software based there will not be a big environmental impact. A website would have to be hosted, the electricity used will be the biggest environmental impact of this. We will also need an office space, where again the electricity is the biggest environmental impact. We will not necessarily be using renewable energy, but instead we will use the cheapest electricity available. We make this choice based on the utilitarian ethical theory. We think the utility of having a product that helps visually impaired children is more important than the utility of using renewable energy.

INTERNAL STRUCTURE OF OUR COMPANY

As a start up company we want to have a moral code to guide us in our decisions. In this section we will describe the important parts of this moral code.

We believe our product can be developed with a limited group of people. Because of this production will take place in the Netherlands, the home country of all members of our project group. As the working conditions are, by law, very good in the Netherlands, we will not have to focus on this too much.

We will have to employ other people at some point and that means we will need an employment policy. We think it is our social responsibility to welcome disabled people as employees. Even more so as we are making a product for disabled people.

You often hear there are not enough women or immigrants in management positions in companies. We think it is not necessary to have a certain percentage of women in management positions. Instead we think it is important to only look at characteristics like dedication and work ethic when choosing someone for a promotion or choosing someone to give a reward to. We came to this decision when we looked at it from Kantian ethics and this quote:

> *Act only according to that maxim by which you can at the same time will that it should become a universal law.*
>
> Immanuel Kant (Groundwork of the Metaphysics of Morals, 1785)

If we would be a woman or an immigrant we would not like to be discriminated. We would like to be evaluated on what we do and not on our gender our skin colour.

CONCLUDING REMARKS

It is usually impossible to satisfy everybody and only do good. That is not what we tried to do with this ethical evaluation. Instead we tried to be realistic and where possible choose for our costumers benefit. New ethical problems will undoubtedly arise in our company, but we can not foresee them at the moment. We hope that this document covers most ethical problems and that it can guide us in new ethical problems.

Bibliography

[1] R. Duba and B. W. Kootte, *Sound localisation in games for visually impaired,* Not yet published.

[2] M. C. Bisschop and J. C. Noortman, *Audio reflections in an audio-based game for visually impaired children,* Not yet published.

[3] *Visio, expertisecentrum voor slechtziende en blinde mensen - blindeninstituut - koninklijke visio,* `http://www.visio.org/nl-nl/home/over-visio`, accessed: 17-06-2015.

[4] D. Gentile, *Pathological video-game use among youth ages 8 to 18 a national study,* Psychological science **20**, 594 (2009).

[5] Z. Cattaneo, T. Vecchi, C. Cornoldi, I. Mammarella, D. Bonino, E. Ricciardi, and P. Pietrini, *Imagery and spatial processes in blindness and visual impairment,* Neuroscience & Biobehavioral Reviews **32**, 1346 (2008), special section: The European Workshop in Imagery and Cognition: Neurocognition and Visual Imagery.

[6] *Audiogames, your resource for audiogames, games for the blind, games for the visually impaired!* `http://www.audiogames.net/list-games/`, accessed: 04-06-2015.

[7] J. Friberg and D. Gärdenfors, *Audio games: New perspectives on game audio,* in *Proceedings of the 2004 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, ACE '04 (ACM, New York, NY, USA, 2004) pp. 148–154.

[8] J. R. Parker and J. Heerema, *Audio interaction in computer mediated games,* Int. J. Comput. Games Technol. **2008**, 1:1 (2008).

[9] S. Targett and M. Fernstrom, *Audio games: Fun for all? all for fun!* (2003).

[10] *Rubber ducky zombie shooter app for the blind,* `https://www.accessibility.nl/projecten-en-publicaties/mobiele-apps/rubber-ducky`, accessed: 04-06-2015.

[11] *A blind legend,* `http://www.ablindlegend.com/`, accessed: 04-06-2015.

[12] *Unity - manual: Audio reverb filter,* `http://docs.unity3d.com/Manual/class-AudioReverbFilter.html`, accessed: 04-06-2015.

[13] *Wii game for visually impaired, blind and fully sighted children,* `http://www.visio.org/nl-nl/wii-game`, accessed: 04-06-2015.

[14] D. Grammenos, A. Savidis, Y. Georgalis, and C. Stephanidis, *Access invaders: Developing a universally accessible action game,* in *Computers Helping People with Special Needs* (Springer, 2006) pp. 388–395.

[15] B. Rakerd and W. M. Hartmann, *Localization of sound in rooms, ii: The effects of a single reflecting surface,* The Journal of the Acoustical Society of America **78**, 524 (1985).

[16] D. M. Howard and J. A. Angus, *Acoustics and Pschoacoustics* (Focal Press, 1996).

[17] J. Middlebrooks and D. Green, *Sound localization by human listeners,* Annual Review of Psychology **42**, 135 (1991), cited By 490.

[18] J. Blauert, *Spatial Hearing* (The MIT Press, Cambridge, Massachusetts, 1983).

[19] B. C. Moore, *An Introduction to the Psychology of Hearing* (Academic Press, 2003).

[20] S. S. Stevens and E. B. Newman, *The localization of actual sources of sound,* The American Journal of Psychology **48**, pp. 297 (1936).

[21] P. R. Singh, *Blind handicapped vs.technology: How do blind people use computers?* (2013).

[22] M. F. Chiang, R. G. Cole, S. Gupta, G. E. Kaiser, and J. B. Starren, *Computer and world wide web accessibility by visually disabled patients: Problems and solutions,* Survey of Ophthalmology **50**, 394 (2005).

[23] E. D. Mynatt, *Transforming graphical interfaces into auditory interfaces,* in *Conference Companion on Human Factors in Computing Systems*, CHI '95 (ACM, New York, NY, USA, 1995) pp. 67–68.

[24] W. W. Gaver, *Chapter 42 - auditory interfaces,* in *Handbook of Human-Computer Interaction (Second Edition)*, edited by M. G. H. K. L. V. Prabhu (North-Holland, Amsterdam, 1997) second edition ed., pp. 1003 – 1041.

[25] *Python (programming language),* `https://www.python.org/about/`, accessed: 08-06-2015.

[26] *Pygame,* `http://www.pygame.org/wiki/about`, accessed: 12-06-2015.

[27] *Freesound.org,* `https://www.freesound.org/home/`, accessed: 17-06-2015.