TH Köln

PROJECT REPORT

Realizing spatial listening tests in Virtual Reality using Unity

Alexander Müller

Supervised by
M.Sc. Melissa Andrea Ramírez Caro
&
Prof. Dr. Christoph Pörschmann

November 19, 2021

Contents

1 Abstract

Central Auditory Processing Disorder (CAPD) is described by the American Speech-Language-Hearing Association (ASHA) as a condition, which "may lead to or be associated with difficulties in higher order language, learning, and communication functions" without being caused by actual hearing loss or inabilities [5]. Focusing on the aspect of spatial hearing Cameron and Dillon established the term Spatial Processing Disorder (SPD) and designed the Listening in Spatialized Noise (LiSN) & Learn auditory training software to improve binaural processing abilities of affected children [1].

Based on this foundation a new training software shall be created as an OpenSource project with Virtual Reality (VR) support inside the *Unity* game development engine. Apart from offering a free and easy to use alternative to the follow up product of the original program (*Soundstorm)* [1] this project shall also evaluate the possible improvements to the concept using the features of an VR application. This includes improved immersion into the auditory environment, the support of real 3D-audio in conjunction with head tracking.

[1] See `https://www.soundstorm.app/`

2 Introduction

Spatial hearing describes the ability, to localize the origin of a given sound event, by using binaural clues of the respective signal. The more obvious advantages of this capability include an aid in orientation and improved possibilities to react to events (like an approaching car), which aren't in the filed of view.

But apart from this, spatial hearing also allows to separate different sound sources and helps managing noisy environments. This is especially useful if the incoming audio signals are similar to each other. A popular example for this is the so called "cocktail-party-effect", which basically described a situation where a listener has to distinguish between a lot of speech signals and focus on a single one in order to be able to maintain a conversation.

Even though this scenario does not cause too much problems for the average person, it contains a lot of problems for everyone with hearing inabilities. Unfortunately, in many cases these impairments affect the ability of spatial hearing and therefore reducing or eliminating information that could otherwise be used to separate the different audio sources from each other. Even modern day hearing aids still can't offer the nuances required to precisely determine the location of a given sound source [4].

But, apart from physiological issues causing problems with spatial hearing, it has been found that children with normal hearing thresholds still might not be able to correctly interpret binaural clues. This effect is described as SPD. Since the actual hearing abilities of the affected children is not impaired, they are - in contrast to persons with typical hearing inabilities - provided with all the information required to localize a given sound source. Based on this knowledge the idea was formed, to treat SPD by *training* affected children and therefore teaching them, how to correctly interpret binaural clues.

Within the *Development and Evaluation of the LiSN & Learn Auditory Training Software for Deficit-Specific Remediation of Binaural Processing Deficits in Children: Preliminary Findings* [**?**] paper by Sharon Cameron and Harvey Dillon it is described how such a training process could look like and already gave some hints [2].

Building upon this approach the training concept described by Cameron and Dillon shall be transferred into a virtual reality application. This shall bring the additional advantage of combining auditory and visual clues and also allows to further extend the process.

3 Motivation

Realizing this listening test as a Unity-based VR application offers several advantages. The first one being the open nature of this project. Since only free-to-use assets have been included, both the compiled application and the source files can be made publicly

[2] The original experiments have only been done with a small sample size. Within the conclusion of the paper the requirement of a clinical trial is mentioned, to validate the efficacy of the training process.

available. Combined with simple setup required to perform the listening experiments, this will hopefully allow a variety of interested groups to perform these tests to collect further data and offer affected children access to work on the condition.

Furthermore the Unity framework in conjunction with the prefabs and scripts, that have been created for this project, it would be relatively easy to extend upon the original training concept. For example the addition of multiple distracters within a given scene could very easily be realized.

Apart from the previously mentioned aspects the usage VR peripherals for this project also offers a lot more possiblilites. Some of them will be discussed in **??**. But especially adaptive 3D audio in combination with head tracking would allow for a much more realistic scenario. Also the novelty of the VR headset itself will most likely already spark a lot of interest in the participants in comparison to a regular learning/training game.

4 Fundamentals

Before describing the general approach of this project in section 5, some underlying principles and theoretical foundations shall be discussed. The contents within this section are not supposed to be used as a general explanation but rather as a brief introduction into the topic.

4.1 Spatial hearing

In order to localize the source of a given noise, we can use several clues within the perceived sound event. The most important ones being the differences in time and level between the signal on the right and the left ear (Interaul Time/Level Differences [ITD/ILD]). Additionally there are spectral colouring effects, which are based on the path an audio signal takes from it's source, around the listeners head and upper body into the ears. In comparison to ITD/ILDs this effect can only be applied to well known noises, because the listener has to compare the currently perceived noise with previous times in order to identify the spectral differences and assign them to a general direction.

The human ability of locating audio sources can be divided into two main features. The first being the perception of the time and level differences between an acoustic event on the left and right ear. The different time points at which the signal is perceived at the two ears, translates to a different phase at which the audio wave is registered. On the other side the level differences mainly derive from shadowing caused by the head. Both of these effects are not frequency independent, for lower tones the localization via phase differences works better, while at higher frequencies the level variations offer a better indication.

The second way of locating a sound source is based upon the spectral differences that occur through the different paths a sound wave can take around the listeners head and upper body. This effect is strongly influenced by the shape of the outer ear, but also the

general geometry of the head makes a difference. Basically the head and the ear can be described as directional filters. This kind of localization is mostly based on experiences. The listener can learn to associate the spectral differences in a known noise to the location from which the noise is originated (e.g. by looking for the sound source).

4.2 Auralization

Auralization describes the process of recreating not only a given audio signal but also the spatial information. With correct auralization a recorded scene (e.g. a classical concert in a concert hall) can be rec
This requires advanced recording processes to capture the spatial information of a scene (e.g. a concert hall)
Auralization describes the process of adding spatial features to audio signals. The simplest example for this would be the move from mono to stereo audio. However, since these initial steps the options of simulating and re-creating spatial features in digital audio processing have advanced quite a lot. This can be easily seen in many consumer electronic products such as AV-Receivers, Soundbars or Headphones supporting 5.1, 7.1 etc. or in particular VR peripherals coming with their own 3D audio frameworks for developers.

The basic principle remains more or less the same for all these products. Within a given *listening space* the original auditory environment shall be recreated.

4.3 Hearing perception (energy and similarity)

4.4 Training abilities

Within this section it shall be discussed, which improvements are to be expected and how they can be explained.

Over the duration of a couple of months children with SPD have been able to improve their Speech Reception Threshold (SRT) up to 3 dB when being provided with binaural clues within the training game. In comparison the results of the control group haven't changed noticeably. This leaves the participants of the study with SRT levels similar to childs with a CAPD.

Since the issue of the disorder is cognitive, it is not too surprising that the issue can be combated with training.

5 Approach

Within this section the general design principles and requirements shall be described, before the drafts for how this could be realized will be discussed in section Concept.

There are three main tasks which shall be used as guidelines for this project:

1. Re-creating the original software to ensure comparability

2. Only using OpenSource / free license assets to allow for free distribution
3. Create all assets and code segments to be easily adjustable for further extensions

Comparability To validate if the new application does actually qualify as a training environment to improve spatial processing abilities, it is sensible to re-create the existing software as close as possible. This would not only allow to compare data from existing research with newly collected data, but also eliminate a lot of effort to check if e.g. the chosen speech stimuli are appropriate for the desired purpose. Additionally if this project does indeed lead to a final application, which could be proven to be a viable alternative, the existing data pools can also be used as reference to evaluate whether extensions (like the addition of head tracking) improves the training effect.

Free distribution Again assuming the outcome of this project will qualify as a valid training software, the opportunity to be able to make it available as an OpenSource project offers several opportunities. The first one being, that combined with the relatively simple hardware requirements, a free to use software might help improve access to an aid against SPD to previously excluded demographics. Additionally a solution, which is more accessible offers the opportunity to gain access to more data, which could be used in further research (e.g. by adding a voluntary option to share progress data within a database). Another aspect would be, that as with any OpenSource project, there is a possibility that other developers or researches use the project as a foundation and extend upon it. This might results in a final product with a widely improved functionality than what would be possible for a single team. However, in any case it is required to only include free to use assets within the implementation to make sure that such a publication won't break any license agreements. So a lot of caution is required when selection third party assets, like sound effects or graphics.

Extensibility Writing flexible software might be more complicated during the initial development, especially when the scope of the project is rather small. However, to allow the easy addition of new features and simultaneously setting the requirements for a successful OpenSource project, a certain level of flexibility within all parts of the project is necessary. Of course within the given development time frame the extend to that this design principle can be fulfilled is limited, but a lot can be achieved even within the concept phase, if considered and prioritised correctly.

Development tools As already mentioned, the major part of the development will be done in Unity with the addition of Visual Studio as IDE. The other major tool is the Oculus VR SDK, which will be used for audio spatialization and of course rendering the visuals to be properly displayed on the given VR peripherals.

5.1 Unity overview

Unity is a development framework, which is mainly used as a engine for video games. However due to the requirements to realism in modern video games, the amount of audio

processing included, is already pretty elaborate. In conjunction with the large user base and support by hardware manufacturer (e.g. Oculus), Unity offers a great starting point for this project.

5.2 Orientation on LiSN project

5.3 Audio assets

As basis for this project the word lists given in the appendix of the LiSN paper [**?**] are used. Since due to the huge variety of options, the only feasible way to construct the sentences is by assembling a sentence through seperate audio files for each word. This however comes with a problem. Simply recording single words and then playing the files one after another strongly alter the speech flow and accentuation, which would normally be present when the sentence is spoken.

To handle this problem, the words won't be recorded separately but instead a subset of the possible sentences from each list will be recorded. Within this subset all words from the list are included within the recordings. Afterwards the recorded sentences will be cut into the individual words and then will be used the assemble randomly generated sequences based on these assets. This should work fine, since the structure of all sentences within any list remains the same. So the speech flow and pronunciation between the sentences should not be too larger. GET SOME REFERENCES FOR THIS ASSUMPTION!

5.4 Visuals

Since this project shall be used to perform research with children as participants the graphical presentation is of great importance. Especially since Virtual Reality (as of today) is in general a quite unfamiliar setting, it is very important to give the user a sense of space within the setting.

Matching audio parameters to visuals The audio scenario shall be based upon free-field sound propagation. To avoid confusion of the listener the environment presented within the VR application should represent this setting. So a closed room would not be a good option, since we inherently expect reflections and reverberation within such a setting. A better approach would be an open field, where the lack of reflexions feels more natural.

Avatars Another important part is the graphical representation of the audio sources. Since we have the visual component given through the VR headset, the sound should not simply come from an invisible sources, but have a origin which the user can identify through the graphics. This however includes another challenge. Of course it would be possible to create humanoid avatars with complex animations - including lip syncing - to convey that the object is indeed active and not just a passive talking rock. However this would take a huge amount of effort to pull of. Instead a different path will be used here, which is also pretty common in budget oriented game design (e.g. indie games). Through abstract avatars and simple animations/movements it is possible to convince the user, that

the object is alive and active, while at the same time requiring a lot less effort to pull of. SOURCE...REFERENCES

5.4.1 Audio Sources

Within the level both interactive and regular audio sources are implemented. All of them use 3D rendering, so that the preceived sound changes based on the position of the player to the object, emitting the sound. The advantage of adding interactive sound sources is that,...

5.5 Training-Game

5.5.1 Audio files

Target sentences are constructed in a way that the co-articulation in between the sentences of a list is similar enough the allow randomly scrambling up sentences will maintaining a natural sounding result.

5.5.2 Game loop

Continuous distracter stories creating a noise with similar characteristics as the target sentences.

Both the target sentences and the distracter stories shall be omitted by actual objects within the game world. This association between audio and visual representation shall create further immersion (FIND REFERENCE!).

Whenever a target sentence was played, four word options will be displayed. (IMAGES OR WORDS?). When the correct result is picked, the selected option turns green and a celebratory sound is played. Otherwise if an incorrect option is selected, it is marked red and a failure sound is played. Alternatively an "uncertain" option is presented.

SNR: For every correct guess, the SNR is decreased by 2.5 dB. For every incorrect guess, SNR is increased by 1.5 dB. At uncertain at first the same sentence is repeated with 1.5 dB higher SNR. At a second consecutive *uncertain* a new sentence will be played, with again increased SNR.

6 Concept

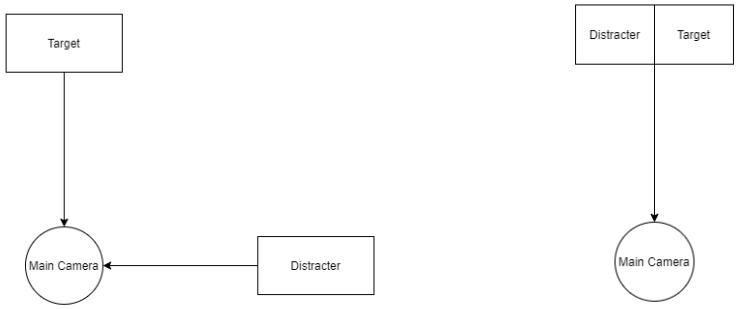
While section 5 only described the general design principles and foundations of the application, this section shall be used to discuss how these requirements actually be fulfilled and which parts of the project might cause issues.

The biggest part will be achieving flexibility in the projects structure, to allow other users to adapt the application to their requirements.

6.1 Training Game

6.1.1 Audio Sources

Inside the training game there will be two possible setup option for the audio sources. In **Setup A** the *target* audio source will be placed directly in front of the *main camera*, while the *distracter* will be moved 90 degrees to the right side [3]. In **Setup B** both audio sources will be placed directly in front of the *main camera* and therefore no spatial information can be used to differentiate both audio streams. In both options the distance of the sources to the *main camera* hjas to be the same.



(a) Setup A (b) Setup B

Figure 1 Training Game Setup

6.1.2 Word databases

Even though the initial scope of the application shall only feature a single word list from the original paper (Appendix A [1]), the implementation should be flexible enough to support different word databases. This is particularly important when considering the option to port the application to different languages. To further elaborate on how this should be done, List 1 from the original paper will be used as an example. Within table 1 all highlighted words are can be used as an *option* to question the user. When observing this table three parameters can be determined:

- Sentences length: **6 words**
- Possibilities for each group: **9 options**
- Number of 'selectable' groups: **3 selectable groups**

Of course this could be extended even further, like adding sentences of variable length or alternating the 'selectable groups'. But as with many of these decisions, offering too many

[3] The selection between right and left side is arbitrary. Interchanging both options during or between sessions would also be an option.

	Subject	Verb	Count	Adjective	Objects
The	**baby**	bought	**two**	big	**apples**
	boy	carried	**three**	blue	**bottles**
	clown	cleaned	**four**	borken	**cars**
	doctor	drew	**five**	green	**chairs**
	girl	dropped	**six**	little	**crayons**
	lady	had	**seven**	old	**cups**
	man	liked	**eight**	orange	**shoes**
	nurse	saw	**nine**	red	**spoons**
	teacher	watched	**ten**	yellow	**trucks**

Table 1 LiSN - List 1

options might lead to problems (e.g. the generated sentences won't be as comparable if they are allow to differ in length).

All three of these parameters have to be considered when creating a dynamic framework, which shall be able to support different word lists. Of course it
Additionally this example also provides and interesting anomaly. All *options* within a given *group* are sorted alphabetically, except the **Count** - words. It is important to not rely on any order in which the list might be created, or any other parameter that depends on the actual content of a given word list.

However, issues like co-articulation effects have to be considered by the creator of a given group.

6.2 Asset management

Moving on from the requirement of establishing a framework, which supports variable *word database* formats the topic of how assets can be added or changed within the application has to be considered.

Unity offers multiple ways to handle this topic. The straight forward approach would be to assign the individual asset files through drag and drop within the inspector. Even though this would in principle fulfil the requirement of changing/adding assets, it's not ideal when many assets shall be changed at once (consider List 1 (table 1, where already 9 x 5 audio files for the individual words would have to be added manually).

Instead the *Resource* system shall be used to handle this group of assets. This offers the opportunity to load assets straight from the file system into the application. However this solution also has some drawbacks. At first the path of the assets within the file system has to be considered. Once option would be to enforce a main path with naming policies. A path could then look like this: *Resources/Audio/WordList<**number**>/Group<**number**>*.

6.3 Progress tracking

In order for this application to be useful, an option has to be added to track the progress of the user over the course of multiple training sessions. At first it has to be defined, which information has to be tracked in order receive all required data to not only monitor possible improvements of a single user but also to compare the results of multiple participants against each other within the context of a study.

The most obvious parameter to be tracked would be the SRT. In order to recognize a progress this has to be stored for each training session. It would of course also be possible to not only keep the average SRT value of each sessions, but also the Signal to Noise Ratio (SNR)s of each individual sentences combined with the information if the guess was *correct*, *incorrect* or *unsure*. However, even though both the target and distracter audio assets have been normalized there might still be some fluctuations within the perceived sound levels of both sources, which can't be tracked or even noise from outside of the application. Therefore this data would come with several uncertainties and is also not required if the data evaluation shall be done in comparison to the original paper.

Another important part would be to allow for differentiation between regular participants of a study and a control group. Therefore a **group** parameter should also be added.

User System Another part which has to be considered, is the option that more than one person might use the same system to do training sessions. To make sure that the progress can be tracked individually, a user system shall be established. Within this system access to the *training game* shall be restricted via a typical login screen requiring the correct **username** and **password** to be entered. Adding new users should also be possible from the login screen, where **username** and **password** can freely be set (as long as the username isn't already in use). Within the 'user creation' progress the already mentioned differentiation between regular and control group could also be added and be linked to the user account.

Progress export Next to tracking the progress data for each user, it should also be made accessible from outside of the application in a format that allows easy evaluation with external tools. For this purpose all user data shall be stored within a *.json* file. This allows to hold multiple types of data from *strings* for the username to arrays of *floats* for the SRT values and easily readable as well as widely used in many different applications.

In-app visualization Finally to application shall give any user an option to keep track of their progress. This will be realized through a separate screen, which features a simple graph plotting the SRT values of the amount of traning sessions done as well as additional space to show e.g. the current average SRT over all sessions.

6.4 User interface

Based on the previously described parts of the application, a general setup for the required User Interface (UI) options can be defined. At first it is important, that all UI elements can easily be selected while using VR peripherals. This could for example be achieved by setting an appropriate minimal size for all text that displayed. Since a lot of this

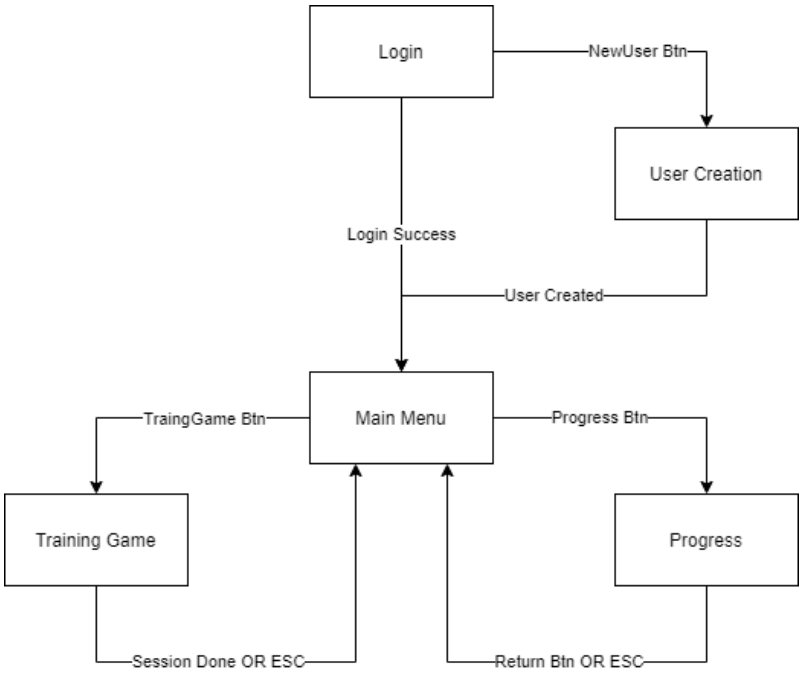
Apart from this general consideration, a brief overview of all major UI elements shall be given

Main Menu The *Main Menu* only requires button to access different screens (e.g. *Training Game* or *Progress*).

Login This requires of course text fields which can be used to type down both the username and password. Additionally a *Submit* and a *Create new user* button are necessary.

User Creation This can basically be the same as the *Login Screen*, with the addition that the user has to select a group (control or regular) to be associated with the account.

Progress As already described, the *Progress* screen shall display a combination of different information. As interactive object a *Return* button has to be added, to allow the user to return to the *Main Menu*.



captionScreens

7 Conclusion

Within part A of this research project, a concept as well as the main set of requirements for recreating the LiSN software has been presented. Of course as within every software

development process it is to be expected that some parts of the final product will derive from the initial concept, due to either new insights into the subject matter or unforeseen difficulties during the implementation.

Abbreviations

API	Application Programming Interface
ASHA	American Speech-Language-Hearing Association
CAPD	Central Auditory Processing Disorder
DAW	Digital Audio Workstation
GUI	Graphical User Interface
HRTF	Head Related Transfer Function
IDE	Integrated Development Environment
LiSN	Listening in Spatialized Noise
SDK	Software Development Kit
SNR	Signal to Noise Ratio
SPD	Spatial Processing Disorder
SRT	Speech Reception Threshold
UI	User Interface
VR	Virtual Reality

List of Figures

Sources

[1] *Development and Evaluation of the LiSN & Learn Auditory Training Software for Deficit-Specific Remediation of Binaural Processing Deficits in Children: Preliminary Findings*, Sharon Camerion, Harvey Dillon, Date: 2011

[2] *Correlating performance on the Listening in Spatialized Noise – Sentences Test (LiSN-S) with the Listening in Spatialized Noise – Universal Test (LiSN-U)*, Kiri Mealingsa, Sharon Camerona and Harvey Dillon, Published: April 2020, Date: 2019

[3] The cocktail-party problem revisited: early processing and selection of multi-talker speech, Adelbert W. Bronkhosrt, Date: April 2015

[4] *Listening through hearing aids affects spatial perception and speech intelligibility in normal-hearing listeners*, Jens Cubick, Jörg M Buchholz, Virginia Best, Mathieu Lavandier, Torsten Dau, Date: 2016, `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6246072/`

[5] *Central Auditory Processing Disorder*, American Speech-Language-Hearing Association (ASHA), `https://www.asha.org/Practice-Portal/Clinical-Topics/Central-Auditory-Processing-Disorder/`