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 28, 2021

Contents

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 [1]. Focusing on the aspect of spatial hearing Cameron and Dillon established the term Spatial Processing Disorder (SPD) and designed the Listening in Spatialized Noise & Learn (LiSN) & Learn auditory training software to improve binaural processing abilities of affected children [3].

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] Reference: `https://www.soundstorm.app/`

1 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 are currently not within the field 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 *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 [2].

Even though this scenario does not cause too much problems for the average person, it contains a lot of difficulties 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 [6].

Next to physiological issues causing problems with spatial hearing, it has been found that children with normal hearing thresholds might still not be able to correctly interpret binaural clues. This condition has been associated to CAPD and was redefined as SPD by Cameron and Dillon in

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* [3] paper by Sharon Cameron and Harvey Dillon it is described how such a training process could look like and already gave some hints [3].

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

[2] For more information refer to [5]

[3] 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.

2 Motivation

Re-creating the LiSN training software as an open-source VR application offers several interesting perspectives and possible advantages. The first point would be the option to establish Unity VR as a valid foundation for listening related training and test projects. If this can indeed be proven, this project can maybe be used to establish a foundation upon more application can be developed with reasonable resources.

The next interesting point would be the evaluation of the additional features that VR can bring into this field like head tracking and a more in depth association between auditory and visual clues.

Moving on, using an open-source and free-license approach also offers a lot of opportunities. Open-source adds the option that other developers or research teams extend upon the base of this project and in turn help creating a more refined product.

Offering the software for free in combination with the - even though not particularly cheap - but rather simple hardware requirements to run such an application, this approach might help improving the availability of such test and training applications for affected persons. If the set-up can be kept simple enough doctors offices, schools or even private households could be able to offer this project.

3 Fundamentals

Before describing the general approach of this project in section 4, 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.

3.1 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).

3.2 Principles of 3D-audio

Where auralization focusses on recreating the acoustic properties of a given environment (e.g. a concert hall or a church), ambisonics describes a procedure of recording and playing back a sound field. Both are important foundations for the establishment of authentic Virtual Acoustic Environments (VAE).

Probably the simplest step towards adding spatial information to audio systems was the move from mono to stereo playback. Nowadays the usage of microphone arrays, extensive speaker setups, Head Related Transfer Function (HRTF)s and potent **DSP!** systems allows for a much more accurate representation of existing and rendering of virtual auditory scenes.

The most common usages of this technologies are most likely surround systems at home or in cinemas and video games. With modern tools it is quite easy to artificially add spatial information to a given noise. To give an example, if a player in a video game shall hear the sound of a door opening behind him, the developers can basically take a "dry" sound (recorded in an acoustically neutral environment) and add reverberation and spatial information later on, depending on the requirements of the scene (distance, size of the room etc.).

In the context of VR environments this is particularly important, because the user is immersed way deeper into the virtual worlds and inconsistencies between the visual and auditory information associated to a noise would be very noticeable.

3.3 Training results

Before re-working the LiSN project, it is sensible to take a look on the results that could be achieved with the original application. Within *Efficacy of the LiSN & Learn auditory training software: randomized blinded controlled study* [4] Cameron and Dillon give an insight on the efficacy of the Listening in Spatialized Noise - Sentences Test (LiSN-S) training program.

Over the course of several studies, it has been proven that auditory training can indeed improve the performance of children affected by SPD significantly. Referring to the work of Cameron and Dillon the LiSN-S training program achieved an average improvement of the participants of 10.9 dB over the course of 12 weeks. This also supports the data from the initial trials during the development of LiSN-S, which found an average improvement of nearly 10 dB over the duration of 3 months [3]. However, in both cases - as mentioned by the authors - the sample group was relatively small, so a level of uncertainty remains.

Apart from simply performing better within the LiSN-S participants showed also improved performances in Fisher's Auditory Performance Checklist[4] [4].

4 Approach

This section will discuss the basic ideas and prerequisites of this project.

4.1 Design principles

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

[4] Reference: `http://www.auditorycenter.com/wp-content/uploads/2014/09/APC-Fishers-Auditory-Checklist.pdf`

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 Integrated Development Environment (IDE). The other major tool is the Oculus VR Software Development Kit (SDK), which will be used for audio spatialization and of course rendering the visuals to be properly displayed on the given VR peripherals.

4.2 Unity overview

Unity is a development platform, which is mainly used as a engine for video games. It is free to use for non-commercial, educational or small revenue purposes and therefore qualifies as a foundation for an open-source project [5]. One of the key advantages of using such a framework is that issues like rendering of graphics and audio, portability between different systems etc. are already taken care of. Also Unity includes a wide variety of features, components and available documentation, which significantly speed up the development process. Also the Unity inspector - if used correctly - allows for several ways of adjusting parameters or switching assets without requiring any programming skills. This is especially interesting for this project, since it allows for a low entrance barrier for third parties who want to alter the application in small means to better fit their requirements. However, the vast scope of options given in Unity also adds some drawbacks, like an increased risk of choosing a suboptimal approach when implementing features.

Oculus SDK The Oculus SDK offers a lot of utilities, which make the implementation of a VR application a lot easier. This goes from prefabs for camera setups, to input management over the a dedicated audio spatializer.

[5] See `https://store.unity.com/#plans-individual` (retrieval date: November 2021)

4.3 Orientation on LiSN & Learn software

Within this section the given parameters for the training game shall be presented, which have to be followed in order to maintain comparability.

- Word list (List 1)
- Target sentence generation (cutting sentences into words, then randomly assembling them)

Game loop Like in the original project the main game loop will consist of target sentences being playing along with a distracter story. After every round **4 options** for a word from the target sentence and an *Unsure* button will be presented to the player. A training session consists of 40 rounds / sentences.

Scenarios The training game shall feature **4 different scenarios**:

1. Same voice with 0 degrees (low cue Speech Reception Threshold (SRT))
2. Same voice with ± 90 degrees (high cue SRT)
3. Different voices with 0 degrees (low cue SRT)
4. Different voices with ± 90 degrees (high cue SRT)

For this purpose both all the target sentences and distracter stories have to be available with two differnt voices (ideally a female and a male one).

Unsure If the player selects *Unsure* the same sentence will be repeated once. If *Unsure* is selected a second time, a new sentence will be played.

Adaptive SNR Based upon whether the participant guess a word correctly (*hit*), incorrectly (*miss)* or select the *unsure* option the audio levels between target and distracter will be adjusted:

- Hit: - 1.5 dB SNR
- Miss: + 2.5 dB SNR
- Unsure: + 1.5 dB SNR

Practice mode To find the SRT of the participant, a *practice mode* will be started at each session, where the given answers are not included in the progress data. A minimum of **5 practice rounds** have to be done before starting the actual training game. Once this minimum has been reached the first incorrect or *Unsure* selection will end *practice mode*. During *practice mode* the Signal to Noise Ratio (SNR) decreases by 3 dB.

Rewards When **5 consecutive correct answers** have been given, the participant is granted a reward sticker, which will be shown on the screen for the remaining duration of the session.

Sound feedback On every *hit* or *miss* either a "success" or "failure" has to be played.

Attention sound 500 ms before the start of every target sentence a short attention sound has to be played (1000 Hz tone of 200 ms).

Playback delays The disctracter story shall always start 2 s before the target sentences and end approx. 1s after the target speaker has finished.

Audio parameters Each target sentence shall be normalized to -22.0 dB RMS. Every distracter story has to be normalized to - 25.0 dB RMS.

4.4 Visuals

Apart from the move to VR peripherals and the new audio assets the visuals will probably be the biggest deviation from the original software. Since the 2D presentation of the LiSN software wouldn't be sensible here, a 3D environment has to be added.

Considering that the target participants of this project are children the visual presentation is of great importance. If the training game isn't engaging, there is a high risk of participants not complying with the required training schedule due to the repetitive nature of the sessions[6].

The novelty of VR in general might already be helpful in mitigating this problem, but it should still be considered during development. Additionally VR applications require special care when it comes the orientation and sense of space within the 3D environment, otherwise there is a risk of players feeling disorientated.

Due to the limited resources in this development, the usage of 3rd-party assets will most likely be inevitable. To still be able to freely distribute the project the license requirement for all non-proprietary assets has to be considered.

Matching visuals to audio Since in the original context neither reflections nor reverberation was included, it shouldn't be added in the re-modelling as well. This adds some conditions to the selection of the 3D environment. A big empty room, would suggest the occurrence of reverb and echo and should therefore be avoided. Ideally an outside location should be selected, where a free-field sound propagation would be natural. Another important point would be the visual representation of the speakers. Since it might be distracting to here a voice from a given direction without being able to associated the sound to any object in the world, the addition of *avatars* for the speakers should be considered.

5 Concept

While section 4 only described the general design principles and foundations of the application, this section shall be used to discuss how these requirements actually be fulfilled

[6] This issue was already brought up by Cameron and Dillon in the conclusion of [4]

and which parts of the project might cause issues.

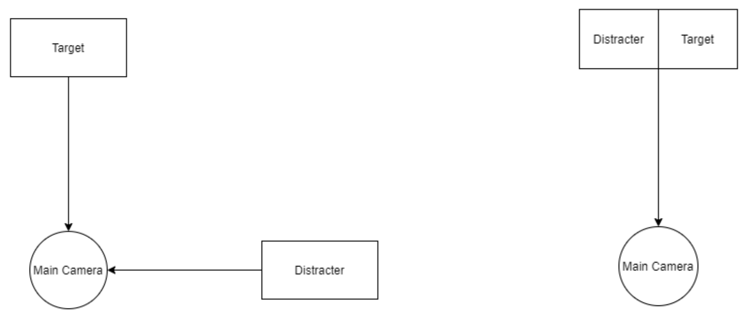
The biggest part will be achieving flexibility in the projects structure, to allow other users to adapt the application to their requirements.

5.1 Training Game

This section will focus on the aspects that are specific to the actual training game. This includes some pre-considerations on how the general setup should look like and what practices can be used to maintain a certain level of adjustability from the start. Of course it is to be expected, that some of the ideas described here won't be implemented exactly.

5.1.1 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 [7]. 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

5.1.2 Word databases

Even though the initial scope of the application shall only feature a single word list from the original paper (Appendix A [3]), 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

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

all highlighted words are can be used as an *option* to question the user. When observing

	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

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 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.

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

5.2 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***>.

5.3 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 SNRs 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.

5.4 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

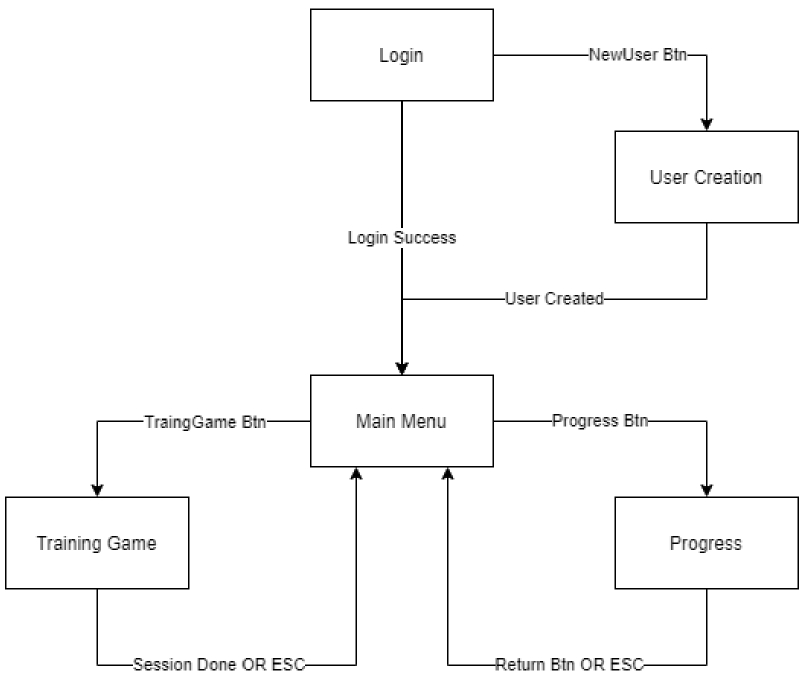
Apart from this general consideration, a brief overview of all major UI elements shall be given

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

6 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

ASHA	American Speech-Language-Hearing Association
CAPD	Central Auditory Processing Disorder
GUI	Graphical User Interface
HRTF	Head Related Transfer Function
IDE	Integrated Development Environment
LiSN	Listening in Spatialized Noise & Learn
LiSN-S	Listening in Spatialized Noise - Sentences Test
RMS	Root Mean Square
SDK	Software Development Kit
SNR	Signal to Noise Ratio
SPD	Spatial Processing Disorder
SRT	Speech Reception Threshold
UI	User Interface
VAE	Virtual Acoustic Environments
VR	Virtual Reality

List of Figures

Sources

[1] *Central Auditory Processing Disorder*, American Speech-Language-Hearing Association (ASHA), retrieval date: Nov. 2021, `https://www.asha.org/Practice-Portal/Clinical-Topics/Central-Auditory-Processing-Disorder/`

[2] *Development and Evaluation of the Listening in Spatialized Noise Test*, Sharon Cameron, Harvey Dillon, Date: March 2006, `https://www.researchgate.net/publication/7326536_Development_and_Evaluation_of_the_Listening_in_Spatialized_Noise_Test`

[3] *Development and Evaluation of the LiSN & Learn Auditory Training Software for Deficit-Specific Remediation of Binaural Processing Deficits in Children: Preliminary Findings*, Sharon Cameron, Harvey Dillon, Date: November 2011, `https://www.researchgate.net/publication/51975813_Development_and_Evaluation_of_the_LiSN_Learn_Auditory_Training_Software_for_Deficit-Specific_Remediation_of_Binaural_Processing_Deficits_in_Children_Preliminary_Findings`

[4] *Efficacy of the LiSN & Learn auditory training software: randomized blinded controlled study*, Sharon Cameron, Helen Glyde and Harvey Dillon, Date: October 2015, `https://www.researchgate.net/publication/276222698_Efficacy_of_the_LiSN_Learn_auditory_training_software_Randomized_blinded_control_study`

[5] The cocktail-party problem revisited: early processing and selection of multi-talker speech, Adelbert W. Bronkhosrt, Date: April 2015, `https://www.researchgate.net/publication/274397643_The_cocktail-party_problem_revisited_early_processing_and_selection_of_multi-talker_speech`

[6] *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/`