Yokohama City University

Bachelor Thesis

Statistical Evaluation of Spatial Attention Shift Caused by Audiovisual Stimuli Using VR

International College of Arts and Sciences
International Arts and Science Studies
Faculty of Sciences
Department of Materials Science
Student ID Number 170451

Author Masaya Tsukida

Supervisor Ruggero Micheletto

Index

Chapter 1 Introduction

1.1 Research Background

Human being always gets stimulation from the outer world using sensory organs such as Eyes and Ears. The function of cognition is crucial to our social life yet is still riddled with mystery. The study of Human cognition and attention has been researched in cognitive psychology for a half-century. In recent years there is a renewal of interest in human cognition because of developing artificial intelligence. However, many things about the structure of cognition remain unexplained. For instance, there is no establishing explanation about how people pay attention to something. Nobody knows what attention is exactly in the first place. It has been challenging to study the function of attention experimental psychologically. It is because human cognition and attention are three-dimensional functions. Recently evolve of electric devices such as VR enables people to experience realistic Vision and Sound. VR means "Virtual Reality."[1] Now, VR is a generic term of visual experience with HMD. In previous research, cognitive science uses a flat display. So, the visual stimulus is limited in the range of 2D. In opposition to that, Using VR devices gets a human experienced 360-degree visual stimulus. Moreover, Using HRTF (Head-Related Transfer Function) technology make it possible for a human to hear 360-degree sound. It is almost impossible to set a stimulus in all directions in the real-world. Another advantage of using VR is the ability to control the stimuli and information the subject receives. In traditional psychology experiments, it is not easy to prepare an ideal experimental environment. For example, for the visual stimuli to appear ten meters away from the subject, an experimental space of ten meters is needed in the Lab room. There is almost no preparing such a vast space with only visual stimuli relevant to the experiment. However, large spaces and equipment can be easily substituted in VR space. That is why the number of using VR in experimental psychology has been gradually increasing.[II] This study focuses on these technologies and thinks that it is good to do a cognitive science experiment in VR.

Let us turn our attention to our society. In Japan, social problems are related to our cognition. People often walk while on the phone. In their blains, the multi-tasks are executed. In another instance, people often do things while listening to music. In their brain, the sound and other tasks are processed. Sometimes these actions lead to a terrible incident. The number of traffic accidents caused by multi-tasking, such as using smartphones in 2019, is 2,645 and increasing in Japan.[III] So, it is natural that unraveling the system of a human attention shift caused by visual and auditory stimuli breaks the ice of solving social problems. For example, unveiling human attention system will contribute to the UI design of devices that stops people from mistaking. Another example is that if attention loss is quantitatively assessed, it may be a steppingstone to regulate using smartphones or earphones while walking. There is much social significance in researching attention.

In this study, the aim is to find out how sound stimulus has an influence on people and the interaction between auditory stimulus and visual stimulus. For the purpose of that, VR experiments are created using Oculus Quest VR and Unity game engine. Moreover, statistical methods are used to analyze the data using Python and R. These approaches are more information scientific than conventional methods. The author majors in science. So, Using the knowledge acquired in the university, this study aims to establish a science-and-technology-based method of cognitive science research and aims to contribute to clarifying the mechanism of human attention.

1.2 Psychological approach to Attention

The selection of sensory input from the outside world and facilitating its processing is called "attention." In particular, the function that enhances the processing ability for sensory input with spatial location information is called "spatial attention".[IV]

This section outlines the function of "attention" with the references.

Attention is known to be classified into three types that depend on its function. First, attention as a function of selecting information is called "selective attention." Next, attention as a function of dividing itself into multiple sources of information is called "divided attention." Finally, attention as a function of keeping attention focused on a task is called "sustained attention." We will discuss "divided attention" in more detail in this study. Divided attention is allocated to multiple tasks and has the concept of "attentional resources," which are required energy for human cognition. There is limited to the amount of available attentional resources.[V] Two theories of it have been proposed. The single resource theory states that all cognitive processing, such as seeing and hearing, uses a single pool of attentional pool, and the multiple resources theory states that attentional resources are used differently depending on the tasks.[VI]

About attention, let us explain another feature of it: The bottom-up and the top-down. Bottom-up is attentional control that is automatically driven by the perceived stimulus, and then attention changes by the characteristics of the stimulus. On the other hand, top-down is goal-oriented and applicable to experiments that use a pre-cue.[VII]

Let us also explain the psychological effects related to this study. First, there is a priming effect. This effect means that the preceding stimulus has an effect on the subsequent stimulus-response.[VIII] Priming effect includes two types of it, direct priming effect and indirect priming effect. It is a direct priming effect in which the primer (preceding stimulus) and target (subsequent stimulus) are the same stimuli, whereas it is an indirect priming effect in which the primer and target are different.[IX] A characteristic of the priming effect is that it works without knowing the relationship between the preceding and the following stimulus.[X]

The other is the Stimulus-Response Compatibility (SRC) effect. This effect is that the response is more accurate and faster when the stimulus and the response share the same characteristics. In the following, we will explain the Simon effect and SNARC effect in the SRC effect based on the paper by Nishimura and Yokosawa.[XI] The participants are presented with a panel on either side of the screen. If the color of the presented panel is red, they are asked to press the right button. Likewise, push the left button if the panel is green. In this case, the red panel on the right side of the screen is responded faster and more accurately than that on the left side. Like these examples, the reaction is faster when the stimulus's spatial features coincide with the spatial features of the response directed by the stimulus. This effect is called the" Simon effect".[XII] Figures 1-2-1 and 1-2-2 show a simpler analogy for the Simon effect based on these examples. SRC effect consists of two features: a facilitation effect when the direction of the stimulus is the same direction as the direction to response, and an interference effect when the stimulus appears in the opposite direction to the response.

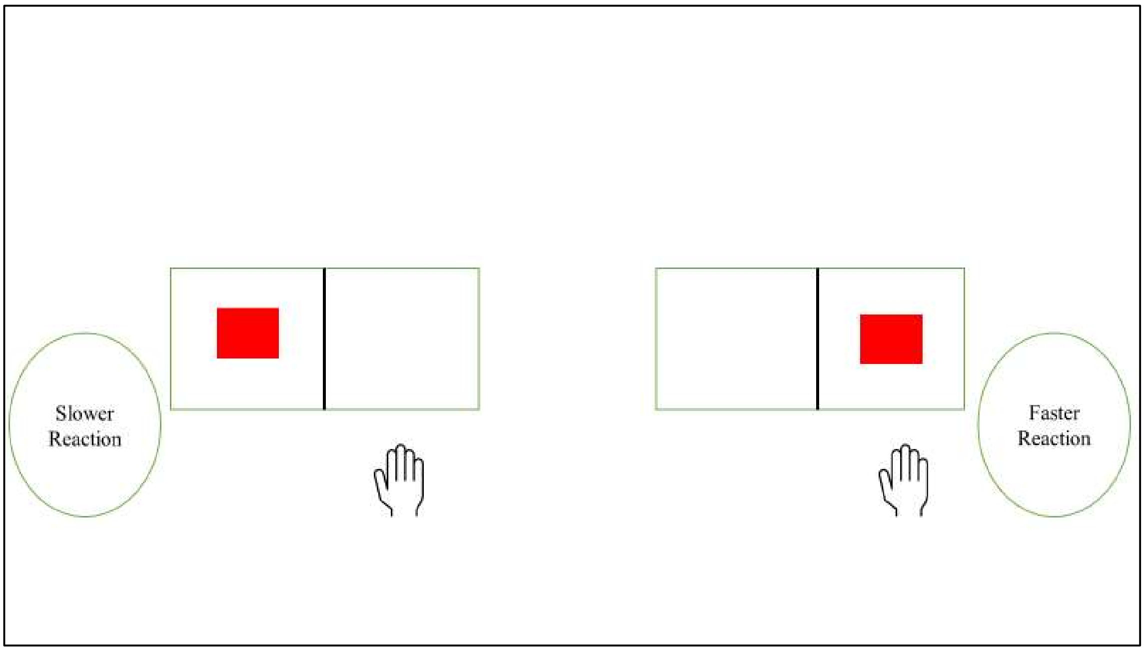


Figure 1-2-1 Examples of the Simon effect. Subjects are instructed to raise their right hand when they see red. In this case, they respond faster if the red color appears on the right side of the screen.

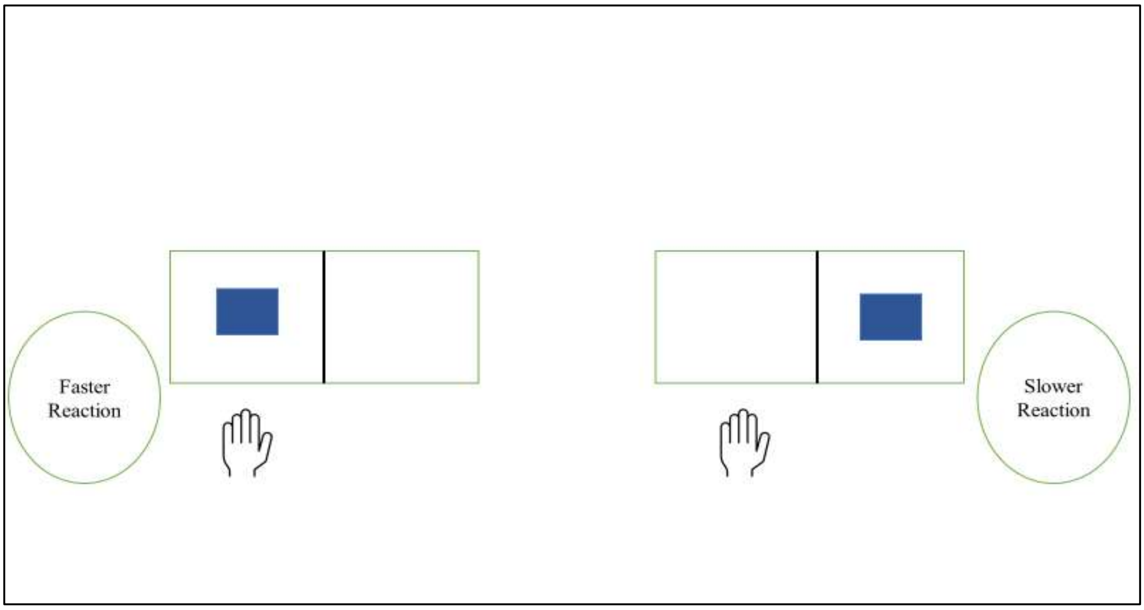


Figure 1-2-2 Examples of the Simon effect. Subjects are instructed to raise their left hand when they see blue. In this case, they respond faster if the blue color appears on the left side of the screen.

SNARC (Spatial-numerical association of response code) is that small numbers such as 1, 2, and 3 are spatially associated with the left side, and large numbers such as 7,8, and 9 are spatially associated with the right side. That is why it is known that reaction to large numbers is faster when stimuli are presented on the right side, and reaction to small numbers is faster when stimuli are presented on the left side.[XIII] This effect is considered to be closely related to the concept of mental number lines.[XIV] In addition to this, the SNARC effect has a connection with eye movement.[XV]

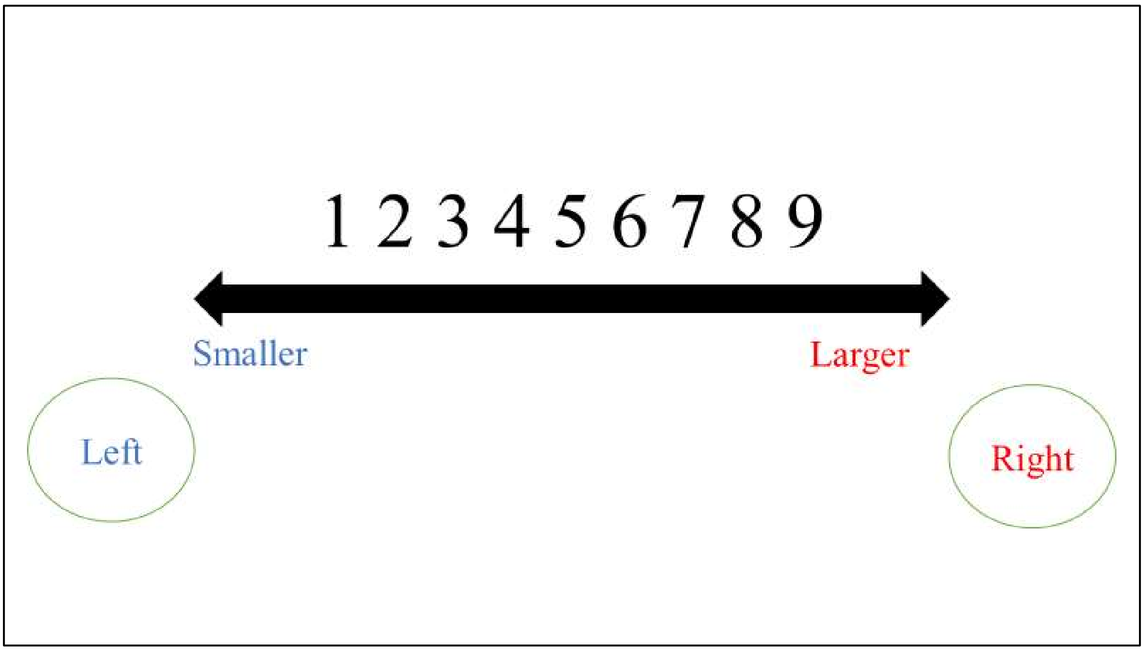


Figure 1-2-3 The relatiomship between mental number line and spatial characteristics.

1.3 Visual system and auditory system of human

This study uses visual stimuli and auditory stimuli. This section describes visual system and auditory system we humans have based on the references.[XVI][XVII]

I. Hearing

Human ears have three-parts such as external ear, middle ear, and inner ear. The external ear contributes to the function that enables humans to locate the spatial location of a sound. The middle ear has a role in impedance matching between the medium of sound and cochlea. The inner ear is consisting of the cochlea, vestibular, and semicircular canals. In the cochlea, hair cells change sound information into neural signals. As we can see so far, the auditory system receives one-dimensional information. Sound localization(echolocation) uses the time difference and level difference between the two ears when the sound is perceived. What is more, the ears lobe and head, which have a complex shape, change the spectrum of the transmitted sound. The brain analyzes these spectral changes to enable sound source localization. This characteristic is called “Head related transfer function (HRTF)”.[XVIII][XIX]

II. Vision

Vision is the most important sensory organ in the human body, and it is said that about 80% of the information we get comes from sight.[XX] The visual system perceives three-dimensional space using binocular disparity and monocular cue from retinal images delivered to both eyes.[XXI] The six principles of Gestalt are known as the principles of visual perception. They are Proximity, Similarity, Common fate, Continuity, Emergence or Closure, Figure-ground. These determine the characteristics of unconscious object perception.[XXII]

III. Multi-sensory integration

A typical example of audiovisual sensory integration is the ventriloquist effect. The ventriloquist effect is an effect in which a sound source seems to be localized at the position of the visual stimulus by simultaneous presentation of visual and auditory stimuli.[XXIII] It has been reported that when left and right visual motion is difficult to detect due to noise, it is easier to see in the same direction as the direction of sound motion.[XXIV] It has also been reported that the reaction time is shorter when multiple sensory modalities are presented simultaneously, not only visual and auditory stimuli. It is called the “redundancy signal effect.” There are two hypotheses, the race model, which is the theory that two stimuli are processed independently, and the coactivation model, which is the theory that two stimuli have interacted with each other and proceed.[XXV]

1.4 Previous Research

This section describes some previous research related to this study.

Vision-Sound interaction: Audiovisual multiple task experiment was carried out (Sakurai and Iwasaki, 2006).[XXVI] This experiment imposed a reaction task to visual ring stimulus with question task, which can be answered with "Yes" "No." The result showed that the difference in the direction of sound stimulus makes the RT difference in the reaction task. What is more, it showed that answering the question makes RT of the reaction task slower. This result suggests that listening to auditory stimuli and answering decreases the source of visual attention. Therefore, visual attention and auditory attention may share the same attentional resources.

For visual-auditory interaction, another one is **the Accessory stimulus effect**. It is reported that Accessory stimulus effects occur in response tasks with visual stimuli, where the simultaneous presentation of auditory stimuli that are unrelated to the target can speed up the response to the task and affect the visual Simon effect. (Fei TIAN and others, 2015)[XXVII]

Eye movements are changed when doing mental arithmetic and reflected spatial biases. (Matthias Hartmann, Fred W Mast, Martin H Fischer,2015)[XXVIII]

Mental addition and subtraction affect spatial attention: Yagi, Nittono, and Shinohara (2017)[XXIX] conducted an experiment that imposed multi-tasks on participants. The first task was to do the addition and subtraction task. Participants answered orally. The second task was to find the target "X" in "N" The result showed that the reaction time was faster when the target showed up upper side under doing the addition and subtraction task. This suggests that the addition task can make human attention to the upper side.

Japanese SNARC effect: there are several theses for the SNARC effect in Japanese. For example, it was confirmed that Japanese people have the SNARC effect of Arabian figures, but the result also showed that Japanese Abacus experts do not have the SNARC effect. These results suggested that it is important that the SNARC effect have a relation to the spatial image of numbers (Nagahara, Hatta 2006). Also, it is reported that the Japanese have a vertical SNARC effect.[XXX]

Cognitive research using VR has not been developed yet. So, there are not many theses. Recently, Harada and Ooyama (2019) conducted cognitive research **using VR**.[XXXI] They show participants 360°VR space. This experiment is that somewhere in VR space, the target showed up. The participants are asked to look for it and push the button when finding it. This is the most recent research about attention with VR space. So. This study used it as a reference to make experiments.

1.5 Unresolved issues in previous studies

In this section, we summarize the unresolved issues in the previous studies described in 1.2, 1.3, and 1.4. First, it should be noted that there are few types of research for Japanese subjects. A few studies have examined the multimodal Simon effect. However, in those experimental designs, the first stimulus is a Simon task, and the second stimulus is more like an accessory stimulus with spatial information (see: Mohammad-Ali, Nikouei Mahani, and others,2019).[XXXII] Even in studies of accessory stimulus effects, there were few studies in which auditory was the primary stimulus and visual was the accessory stimulus. Besides, most of the experiments are conducted using PC monitor screens, and few studies deal with attention in three-dimensional space such as VR. In the Simon task, where the stimulus's spatial location affects the reaction time, an experimental design in which the stimulus is limited to the internal of the monitor may be inappropriate. This is also a problem for reaction time research in general; many studies do not sufficiently examine statistical methods when analyzing data. For example, ANOVA and t-tests are sometimes used for reaction time data that are not normally distributed.

1.6 The aim of this study

This study focuses on the attention shift caused by visual stimulus and auditory stimulus for Japanese speakers. In particular, how the spatial localization of audiovisual stimuli affects reaction time and how auditory perception affects visual perception are the main purpose of this paper. The following five questions are based on the goal of clarifying these two.

I. Does the Simon effect show up by auditory stimulus? (Experiment 1)

II. Visual sense or Auditory sense, which one takes priority in Human cognition? (Experiment 1,3)

III. How is the vision-hearing interaction effect? (Experiment1,3)

IV. Under multiple tasks, does auditory stimulus affect the other modality task? (Experiment 3)

V. For the same content stimulus, is there a difference in physical reaction, such as eye movement, between when perceiving auditory stimuli and When perceiving visual stimuli? (Experiment 2)

To solve these questions will contribute to making UI design such that the human easily reacts to and overall society. That is more, revealing them should lead to an accident-free society. Three new measures are adopted in this study. First, the standalone VR device is used to realize the 3D experiment space. This space enables to show directly visual stimuli and auditory stimuli from all directions (fig. 1-6-1). Second, to make experiments that have much flexibility, the Unity game engine is used. By using it, unique experiments such as 3d visual search tasks are made. Finally, statistical methods are used to discuss the result. I will repeat this —the author majors in science. Hence, the knowledge about physics and mathematics can apply to cognitive science problems. We believe it is very meaningful to study cognitive science and cognitive psychology from a scientific approach.

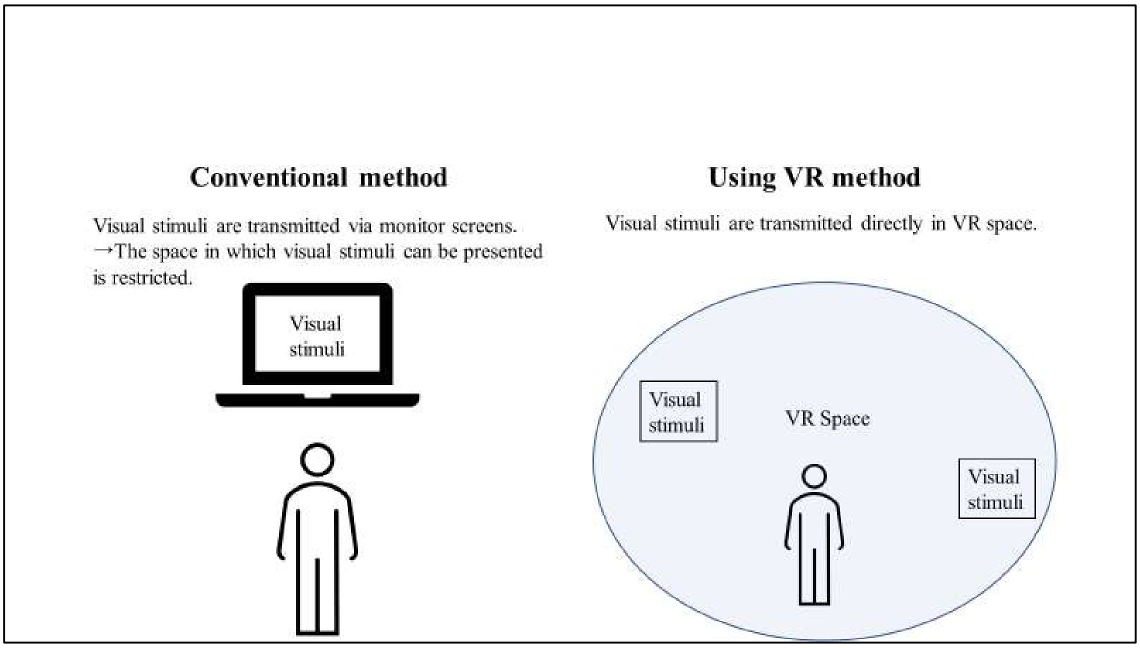


Figure 1-6-1 Differences between the previous experiment and this experiment.

Chapter 2 Experimental method

2.1 About VR

Now, there are many VR devices on the market. They vary from relatively inexpensive to high-priced. The features and performance of the device also vary. The trend of the world is becoming more affordable to buy VR. Also, Stand-alone VR, which does not require a computer to use, has recently been released. Therefore, it should become more popular and will attract a major focus as a research tool. This section describes the VR device which this study used.

We used Oculus Quest made by the American corporation Oculus. This is a standalone VR device. Therefore, it can be used on its own, without the need for an external device such as a PC. We can also experience VR without a cable, without the need for an external camera. It has internal storage, which stores the experiment data in the Oculus. The SoC is Snapdragon 835. 6DOF tracking sensor is built-in. The display resolution of it is 1440×1600, and the refresh rate is 72Hz. Oculus Quest has built-in speakers, but in order to arrange a more accurate auditory situation, this study uses headset Steel Series Arctis7. Participants' responses to stimuli are input by the oculus touch controller.

While the Oculus Quest can be used by itself, there are a few things to set up for first-time use. Using Oculus, the following procedure needs doing. First, pair it up on the smartphone app that makes Oculus complete the Wi-Fi settings and pair it to the Oculus Touch controller. Next, check the developer mode. Finally, connect to the PC via USB-C and allow access to the Oculus's internal storage.

The experiment app needs to be built in Unity and built as an Android app because Oculus Quest works with Android OS. After the app is built, it will be added to the unknown developer section in Oculus Quest.

Another feature of Oculus Quest is a mirroring function. This is the function to show the participant's view on the monitor. The function to supervise Participants' perspectives in the VR space leads to the experiment's smooth conduct. Chromecast is needed to use, and a smartphone that installs the Oculus app, Chromecast, and Oculus must be connected to the same wireless LAN.

Oculus Quest has an accelerometer and an angular acceleration sensor. They can track player's head positions (x, y, z) and head angles (φ, θ, Ψ). In addition to this, Oculus can track the motion of player hands, but this study did not use it.

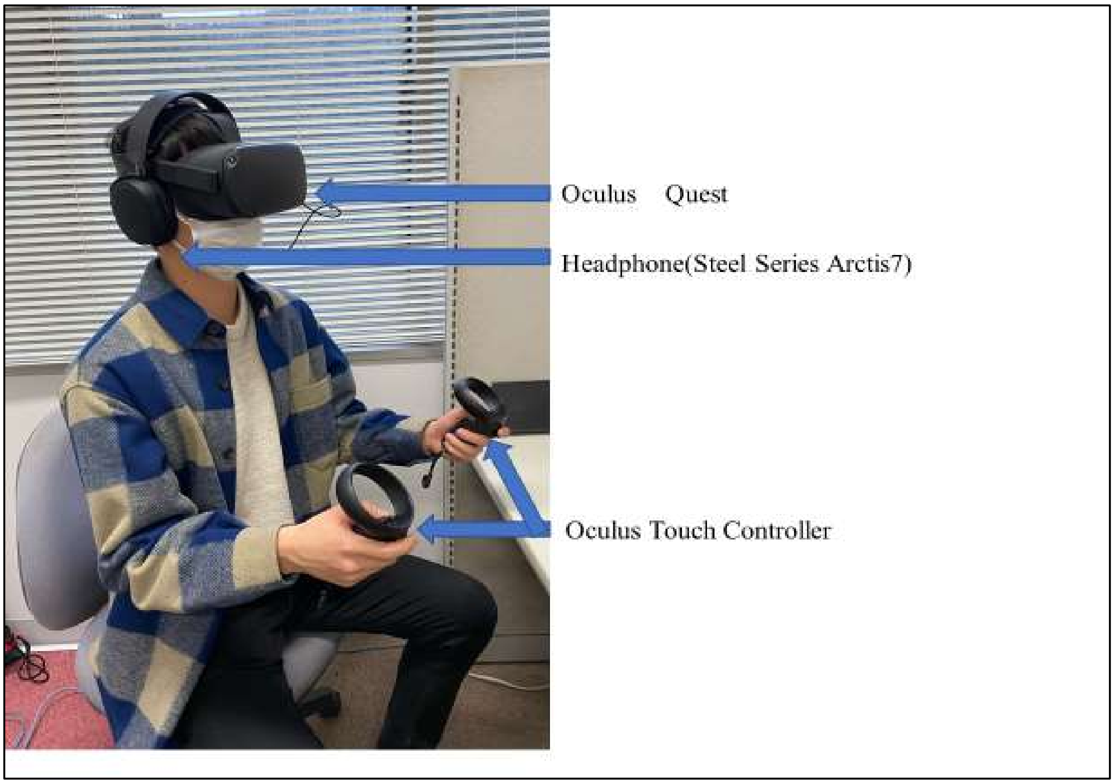


Figure 2-1-1. Experimental environment on experiment 1, 3.

This study also used another VR device called "FOVE" in experiment 2. FOVE is the world's first eye-tracking VR and used by companies and institutions around the world.[XXXIII] FOVE uses orientation tracking IME and infrared position tracking for head tracking. It also uses an infrared eye-tracking system for eye tracking, which has an accuracy of 1.15 degrees.[XXXIV] FOVE eye tracker can get not only Eye-position and orientation but also Eye torsion and whether the user is doing a saccade.[XXXV] The difference between FOVE and Oculus is that it requires a camera for positional tracking and a high-performance desktop computer. Due to the high specification requirements, it has a variety of features such as 120 FPS support, eye movement tracking, and eye image acquisition. Therefore, it is often used in academic applications among VR devices.

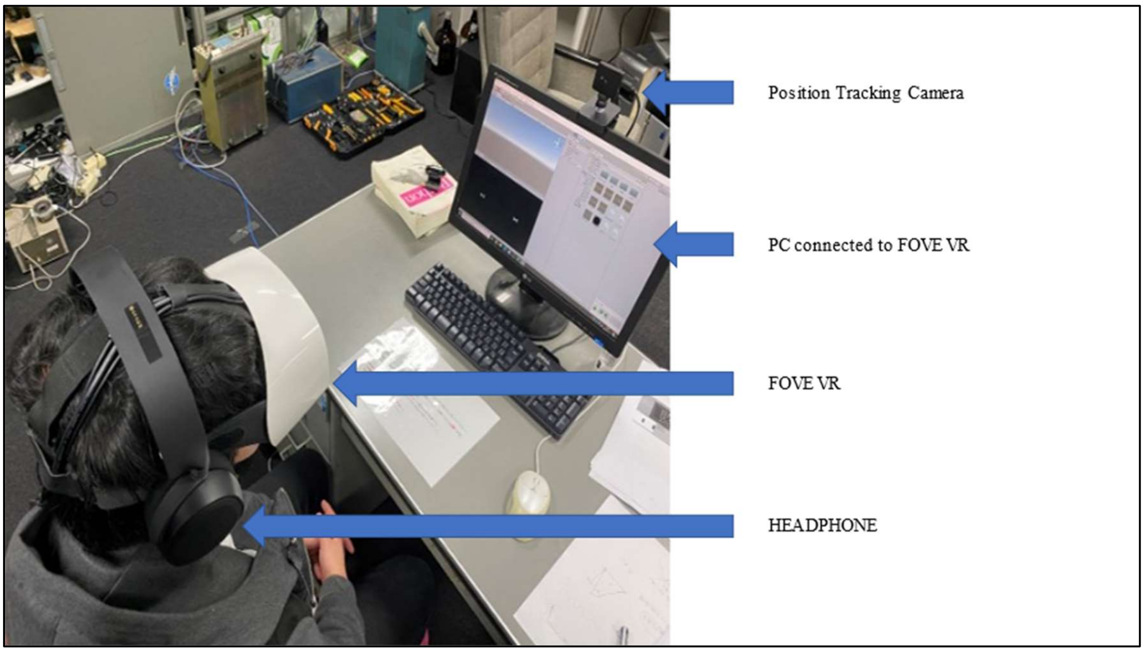


Figure 2-1-2 Experimental environment on experiment 2.

2.2 About Unity

In this study, Using the Unity game engine, the VR space and experiment are made. Unity is compatible with various platforms, and more than 60% of VR/AR content has been using Unity.[XXXVI] Unity is a flexible game engine and versatile, but it is still rarely used in the academic research field. Hence, conducting perceptual experiments with Unity is worthwhile. Scripting in Unity can be used using the C# language. Furthermore, the UI and tools help us build experiments visually. In addition, Unity assets are available for Oculus and steam audio, which is easy to install. For example, when using Oculus Quest, it is easy to use because of the Oculus integration asset.

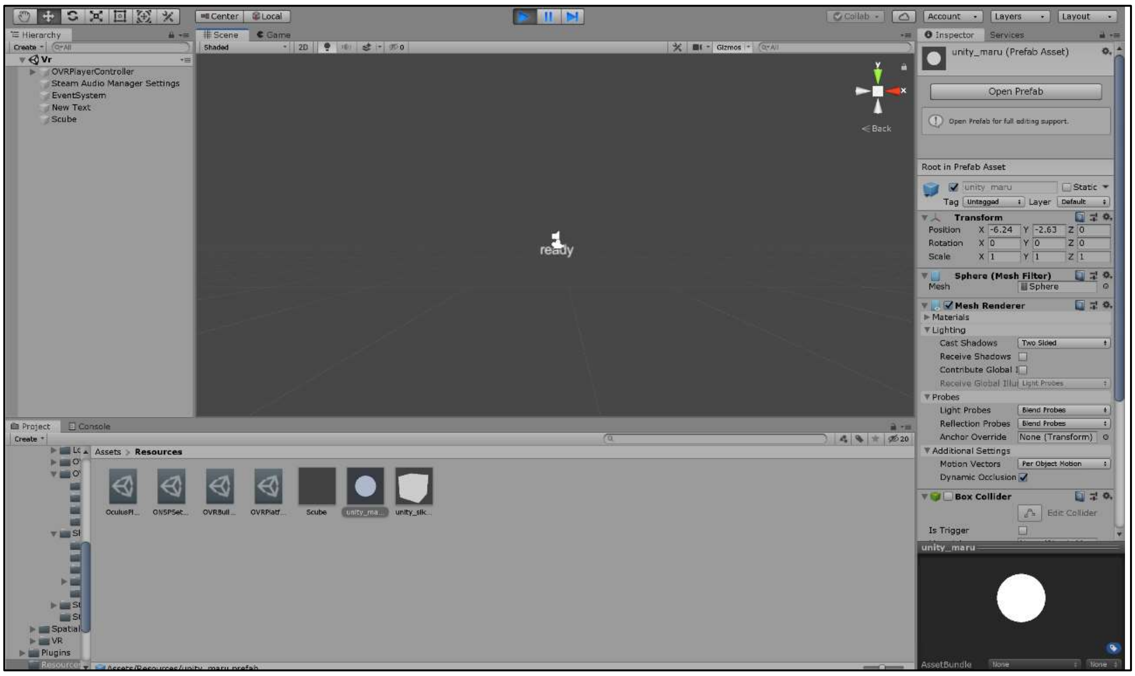


Figure 2-2-1 Unity project run screen.

2.3 About Steam Audio

This study uses a technology called Steam Audio to provide auditory stimulation. This enables to realize immersive audio(three-dimensional-sound) by using HRTF (Head Related Transfer Function) and physical model in the game space created by Unity. Conventionally, auditory stimulation with headphones is channel-based and localized in the head. Steam-audio can provide object-based, Out-of-head localized auditory stimulation. This study used this technology to place a virtual sound source in the game space because the target of this study is the effect of direction of sound and interaction between the sense of vision and sense of sound. In the Simon effect experiment, the auditory and visual stimuli have identical meanings (e.g., the sound of the word "red" and the visual stimulus whose color is red). That is why We arranged visual and auditory stimuli presented from the same spatial position in VR space. In this way, the spatial features of visual stimuli can be matched to the spatial features of auditory stimuli. Note that the HRTF used in this study is generic data.

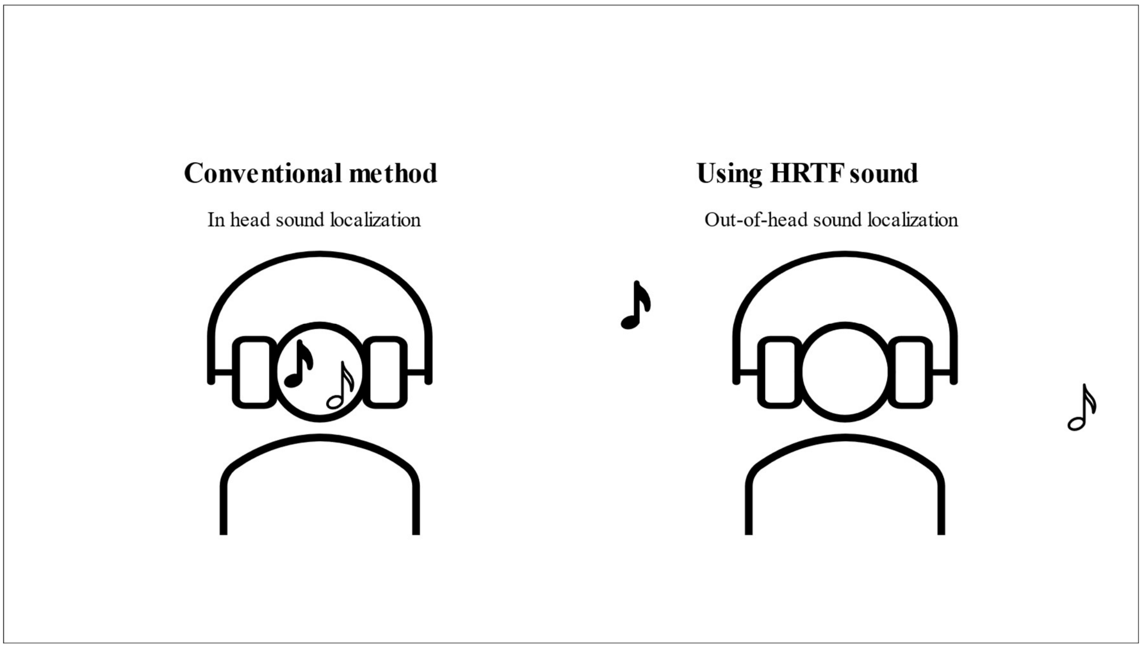


Figure 2-3-1 The difference between raw sound and HRTF sound.

2.4 About R and Python

Python is an interpreter type programming language used for science and technology domain such as artificial intelligence. This study uses Jupyter-notebook, which enable to perform sequential programming and execution. It can also make notes and save the output. There are many libraries of scientific computing and data visualization in Python. So, Python was used to make plots and do statistics tests. And it was also used to make sound files using library pydub.

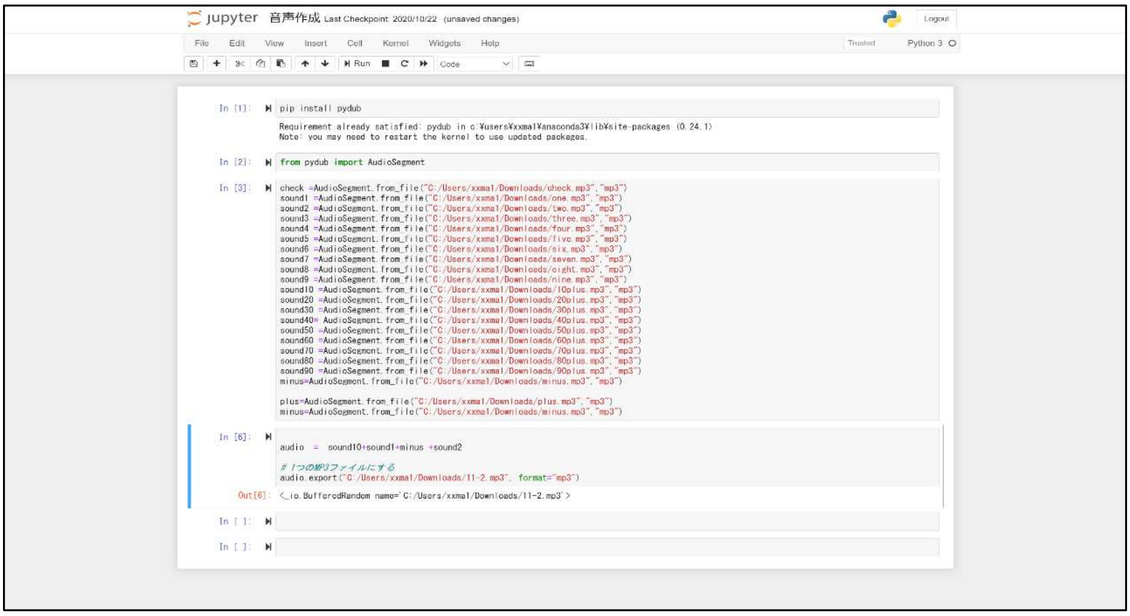


Figure 2-4-1 Jupyter Notebook run screen.

R is a programming language used for researchers and data-scientists in both academic and industry. R is developed for numerical analysis. Therefore, R can easily do statistical analysis methods such as linear regression analysis, decision tree analysis, and data visualization.

Additionally, R has a library for Reaction Time analysis.

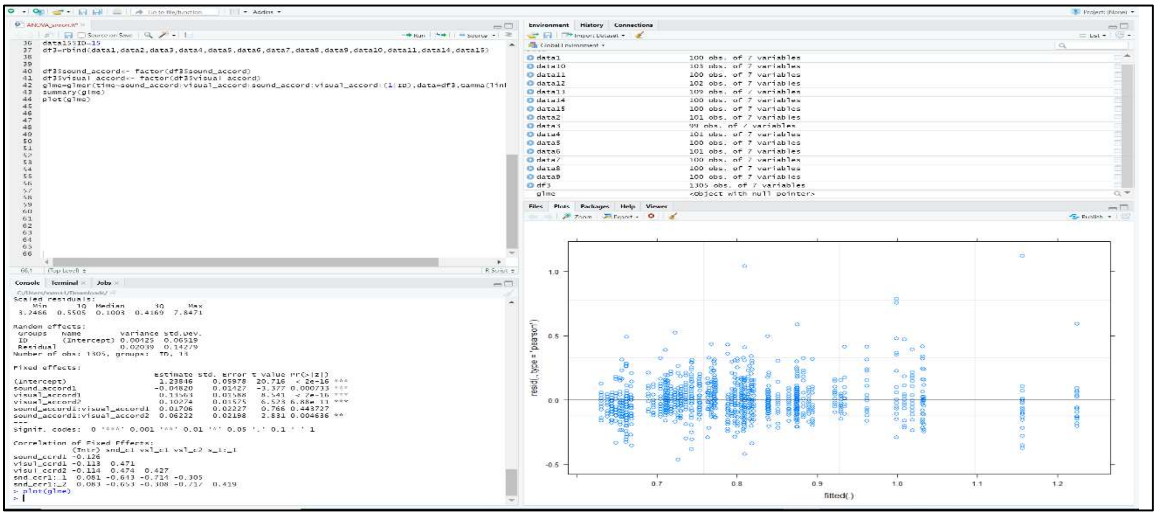


Figure 2-4-2 R studio run screen.

2.5 Procedure of experiments

This section describes the experimental method. There are three main experiments.

In the Visual-Auditory Simon task (Experiment 1), we studied whether the auditory stimuli produce the Simon effect. At the same time, the Visual -Auditory interaction effect was also studied. If the auditory stimuli' spatial location affects the reaction time, the Simon effect is considered to have occurred. We also evaluated the effect of the simultaneous presentation of audiovisual stimuli and the spatial features' consistency on reaction time.

In Experiment 1, we researched the effects of audiovisual stimuli' spatial location on response and attention. Experiment 2, 3, on the other hand, focused on the effects of the mental representations of the stimuli (non-spatial auditory functions effect on spatial visual attention.) rather than their spatial location.

The Audiovisual mental arithmetic task (Experiment 2) aimed to evaluate whether there was a difference in eye-movement between while doing a visually showed arithmetic task and while doing an auditorily showed arithmetic task. If there was a change in gaze when an auditory stimulus is presented, we could assume that the auditory stimulus influences the visual sense. We also examined whether there are differences in eye movements due to SNARC-related effects between addition and subtraction. Eye movement changes during mental arithmetic have been reported in the past using visual and auditory stimuli. We thought that by using two modalities in the same experiment, we could examine whether numerical processing by mental arithmetic affects spatial attention.

In Experiment 3, based on the results of Experiment 2, we investigated whether two tasks can influence each other even if they are different tasks in different modalities under multi-tasking. The Audiovisual multi-tasking task (Experiment 3) aimed to evaluate how the auditory stimuli, such as auditorily presented addition and subtraction problems, affect the leading visual search task. Subjects were asked to perform visual search in VR space. We prepared three types of blocks: one for visual search only, and the other for visual search plus a mental arithmetic (addition or subtraction) task in which the question is shown auditorily. Evaluate the effect of separate auditory tasks on the performance of the primary visual search task.

Reaction times were used to evaluate Experiment 1,3, and Head tracking data was used for Experiment 3. Eye-tracking was used in Experiment2. Experiment 1,3 was made with Unity for Android OS and Oculus Quest. Experiment 2 was made for Fove VR with Windows OS. All experiments were conducted in a 3D VR space. In experiment 1, three-dimensional virtual sound was used by steam audio to create a VR space that corresponded to the auditory space and the visual space (figure 2-5-1).

All experiments were designed so that everyone would undergo trials under all conditions. It is called "within-subjects design." Each experiment was pilot tested against several people, and the experimental design was modified. The results of the pilot tests were excluded from the analysis.

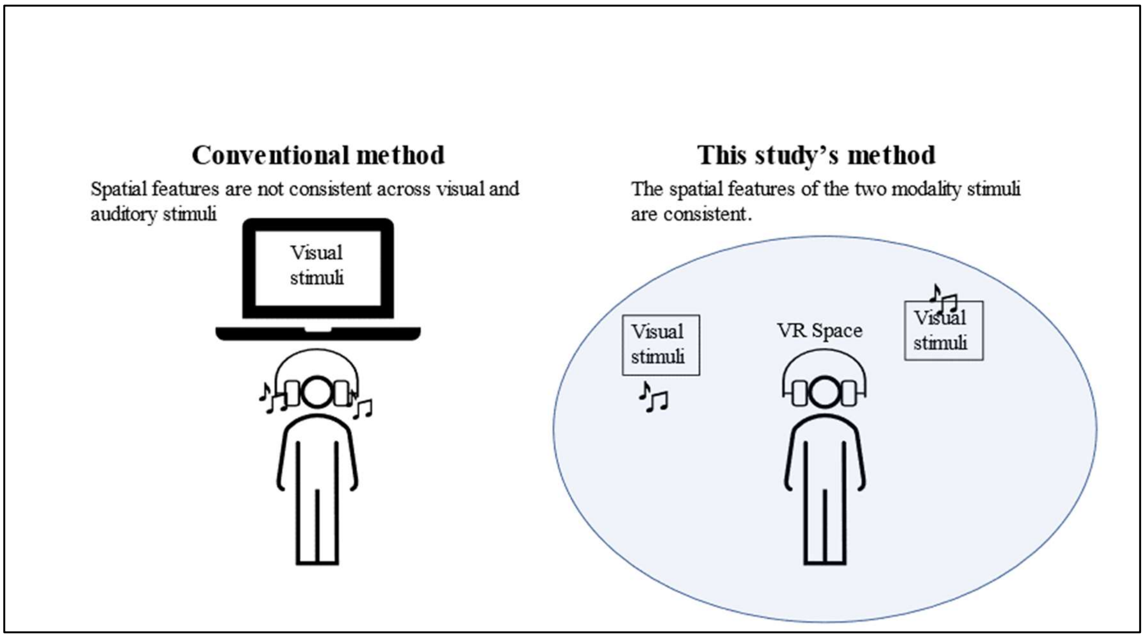


Figure 2-5-1 The difference between conventional method and this study's method.

I. Auditory and Visual Simon task (Experiment 1)

Research Question: Is there an auditory Simon effect? How is the audiovisual interaction effect?

The participants were 15 men and women over the age of 18. They did experiment with a seated state. The analysis used the data of 13 of them who were able to perform the experiment correctly.

I. Display the ready screen (3000 ms)
II. With the display of "start," the auditory stimulus[1] such as the Japanese word "AKA" (means "red") and the Japanese word "AO" (means blue) was played back from the right or the left side of the headphone (4000 ms)
III. Simultaneously, the visual stimulus, whose color was the same as the auditory stimulus, was presented on the left or the right side of the participant's view in the VR space at a certain probability.
IV. Participants were asked to push the controller of the indicated side by the auditory stimuli. When the auditory stimulus "AKA" was presented, they are asked to push the right button. On the other hand, when the auditory stimulus "AO" was presented, they are asked to push the left button. Subjects were asked to follow the instructions of the auditory stimuli, not the visual ones.
V. The sequence of I-IV was repeated one hundred times.
VI. The reaction time from the presentation of the stimulus to the controller input, the direction of the auditory stimulus presented, the presence (and direction) or absence of the visual stimulus, and the correctness were recorded.

---

[1] Auditory stimuli were made with the textalk.

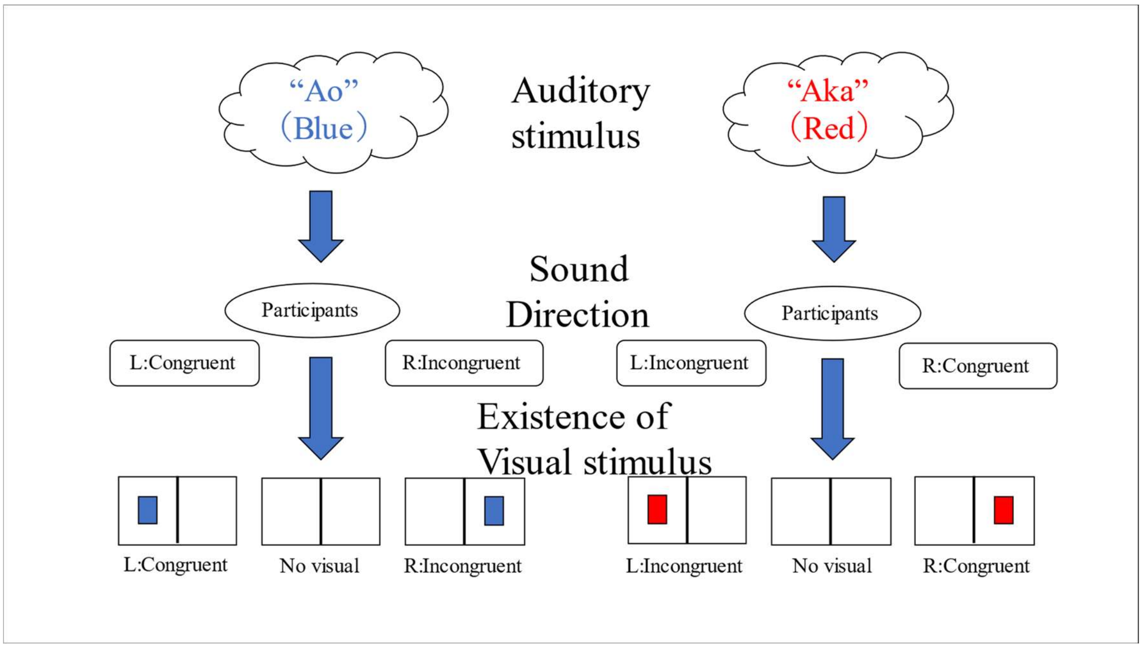


Figure 2-5-2 The structure of the stimuli perceived by the participants.

(visual and auditory stimuli are presented simultaneously)

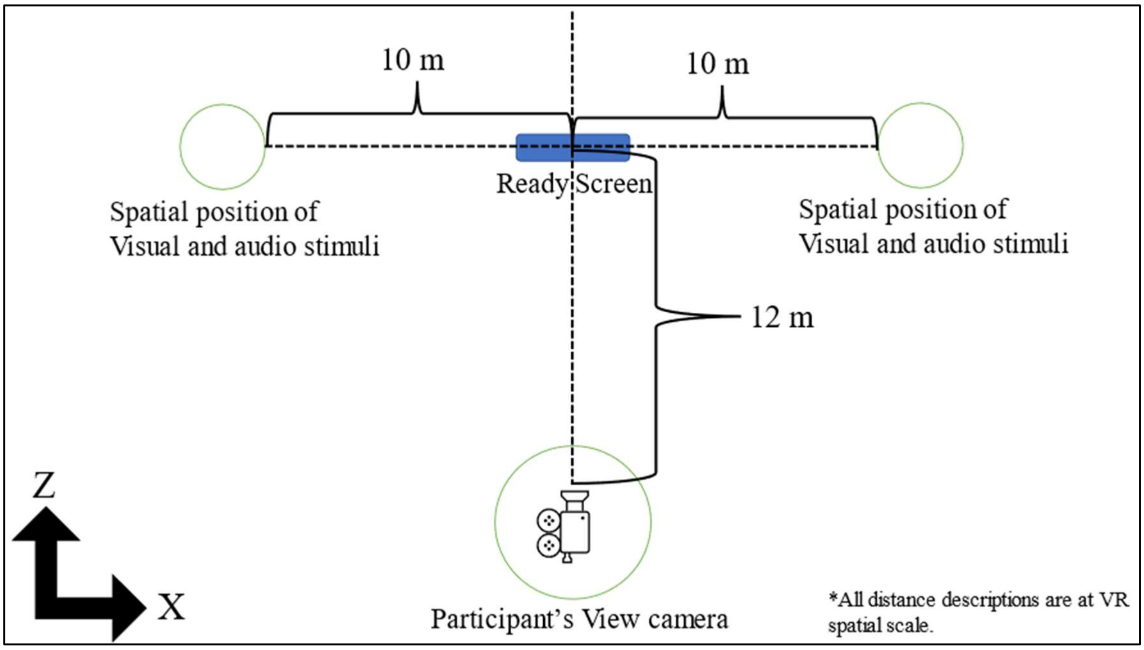


Figure2-5-3 Experiment environment in VR space.

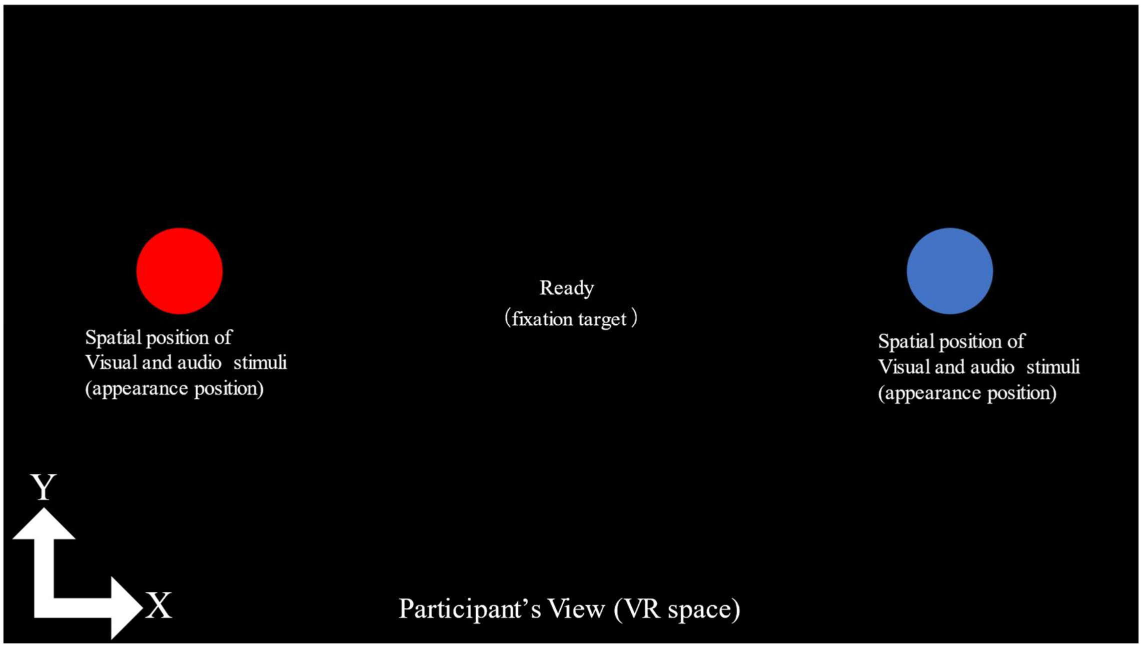


Figure2-5-4 Experiment environment in VR space. Audiovisual stimuli are presented when ready is changed to start.

II. Audiovisual mental arithmetic task (Experiment2)

Research Question: Are there differences in eye-movement between while calculating mentally auditorily presented arithmetic problems and while calculating mentally visually presented arithmetic problems? Does mental arithmetic affect the visual function of gaze?

The participants were 12 men and women over the age of 18. They did experiment with a seated state. The result of two of them notes that the camera coordinates in VR space are variable depending on the spatial position of the head.

I. Presentation of the warning signals (3100 ms)

II. Auditorily presented in Japanese arithmetic problems[2] or visually presented arithmetic problems with Arabic figures were showed (7000 ms)

III. Participants were asked to calculate mentally and answer orally.

IV. The sequence of I-III was repeated 84times. The first four times were the practice block. After them, visually addition, visually subtraction, Auditory Addition, Auditory subtraction were randomly presented.

V. Participant's head-tracking data and Eye-tracking data during the experiment, along with the type of stimulus, were recorded.

---

[2] The Japanese voice data of arithmetic problems used in the experiment was taken from the following website and edited in Python. https://soundeffect-lab.info/sound/voice/info-lady1.html

This audio was monaural and had no spatial location information.

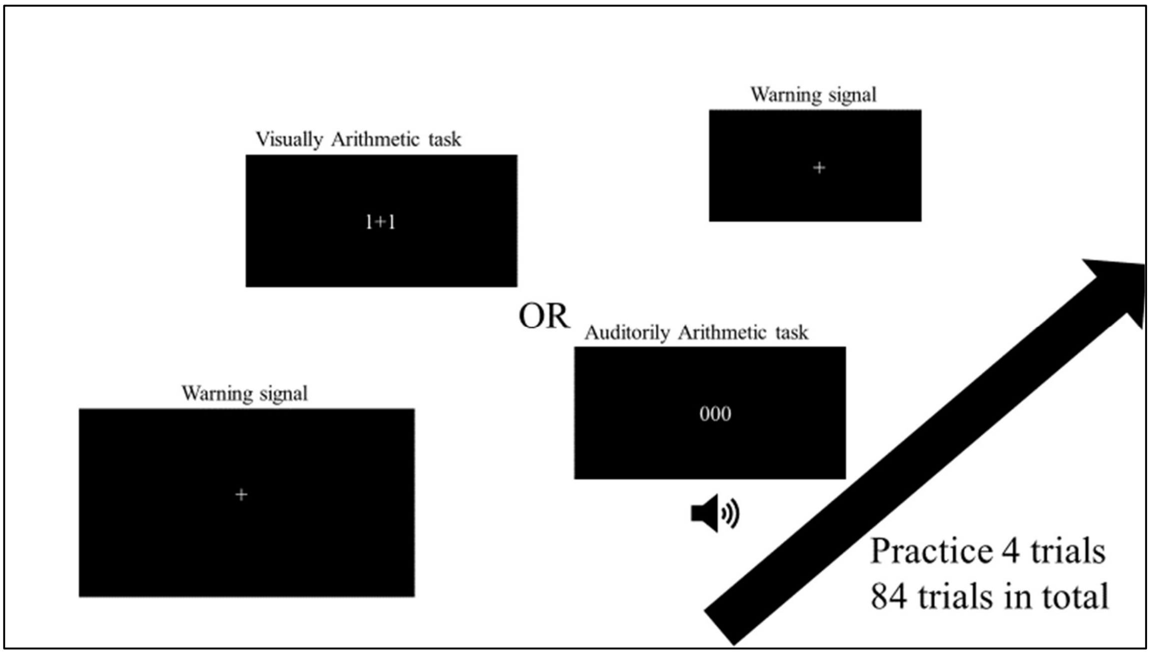


Figure2-5-5 Procedure of experiment 2.

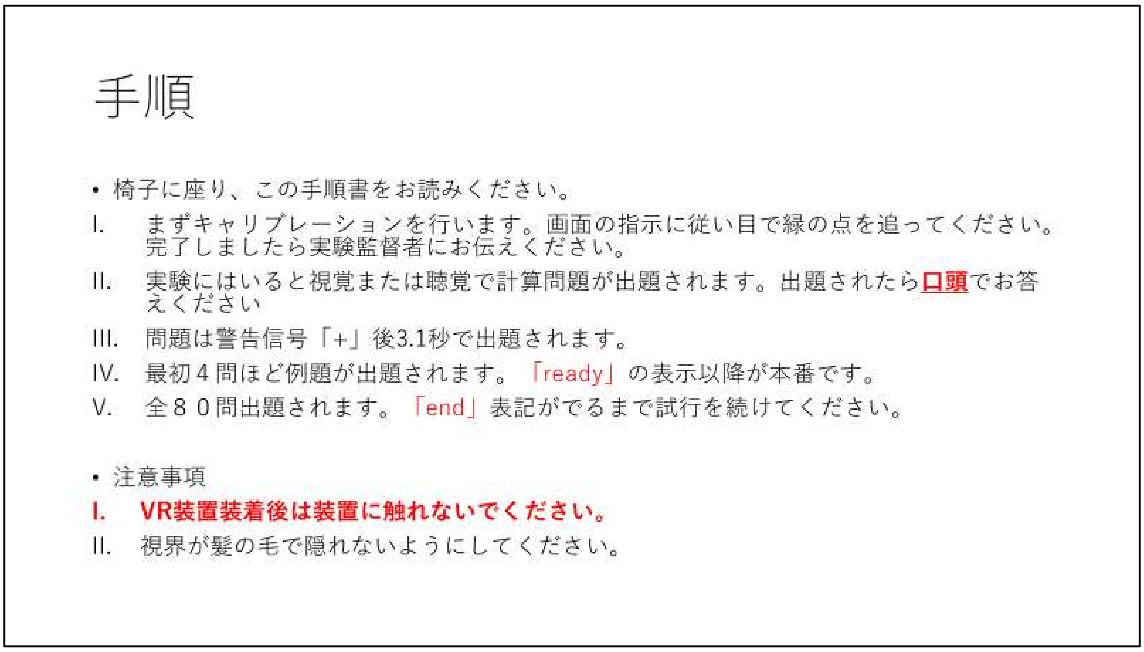


Figure2-5-6 Procedural instruction written in Japanese to participants in experiment 2.

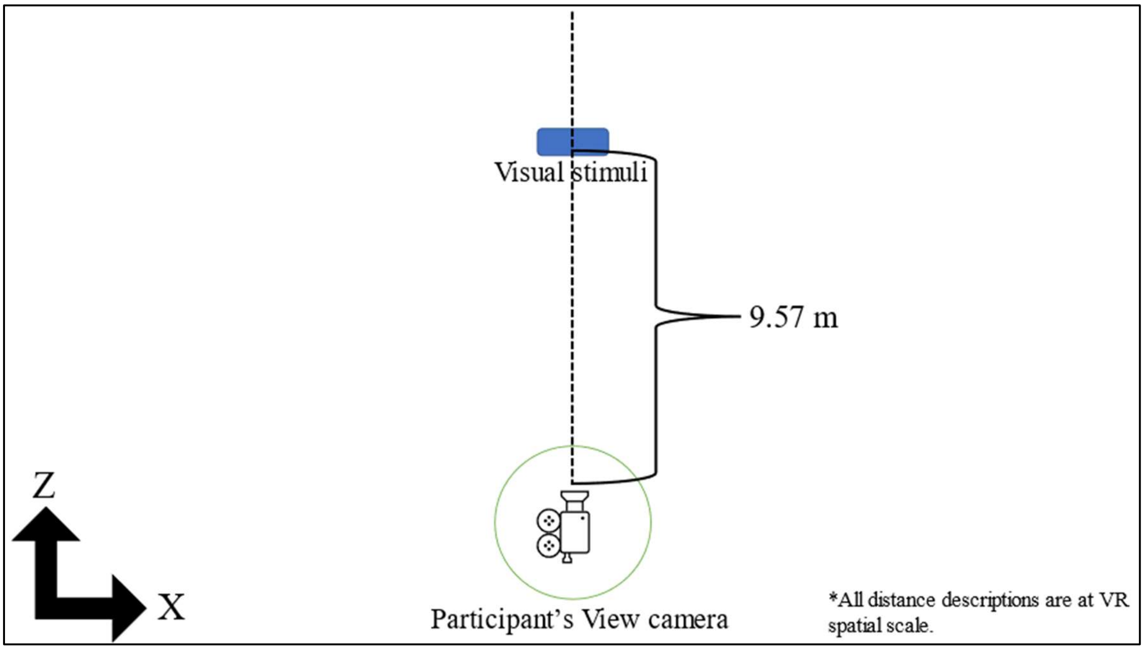


Figure 2-5-7 Experiment environment of Experiment 2 in VR space.

III. Audiovisual multi-tasking task (Experiment 3)

Research Question: How does performance change when separate visual and auditory tasks are imposed? Are the attentional resources used by the visual and auditory systems the same?

The participants were 24 men and women over the age of 18. The experiment was conducted in an upright position. Twelve of them were given the No sound block, addition block, and subtraction block in that order. The others were given the No sound block, subtraction block, and addition block in that order.

I. Presentation of ready screen(5000 ms)

II. Numerous white spheres appeared at equal intervals in front of the participant at 180 degrees in VR space.

III. Participants looked for a white cube (target) among the white spheres and pressed the A button on the controller immediately after finding it.

IV. The white cube (target) vanished, and later another white cube (target) appeared again somewhere in VR space.

V. Participants repeated tasks III-IV continuously until the mark "end" appears (72 trials x 4 blocks, the first 72 trials are practice. Three blocks were "addition task block," subtraction task block," and "No sound blocks." The block order was counterbalanced between participants.)

VI. In some blocks, calculation problems were presented auditorily[3] at the same time. Participants were also asked to calculate mentally and answer orally.

VII. Search time (reaction time), head tracking data, and Target location were measured and recorded.

---

[3] Auditory stimuli are the same as the one used in Experiment 2.

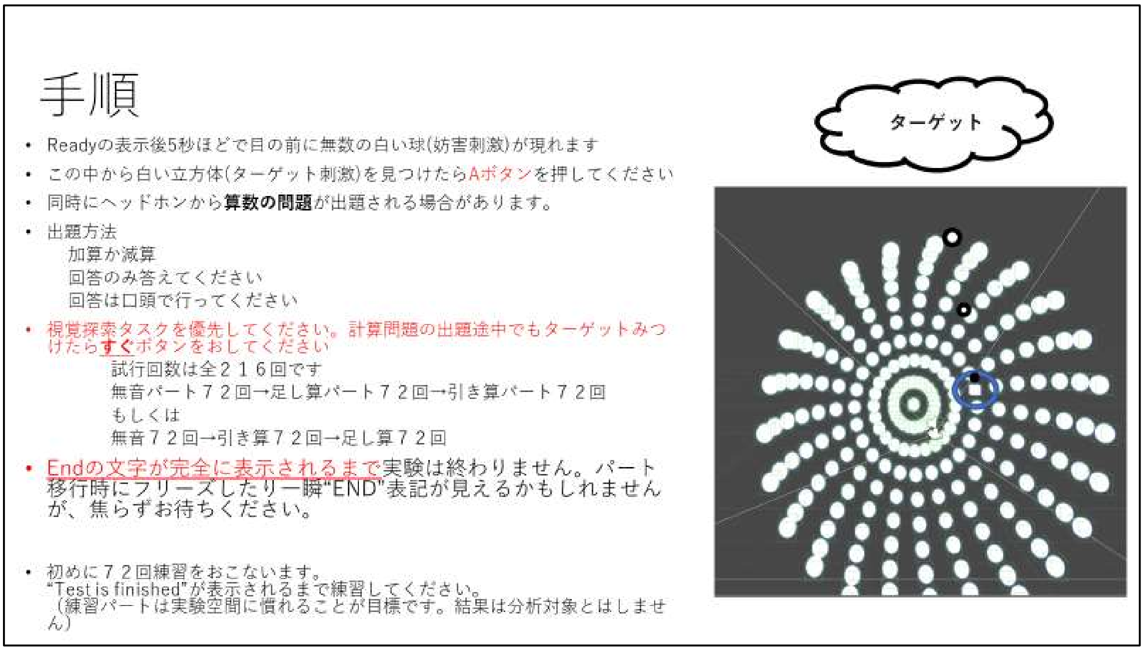


Figure 2-5-8 Procedural instruction written in Japanese to participants in experiment.

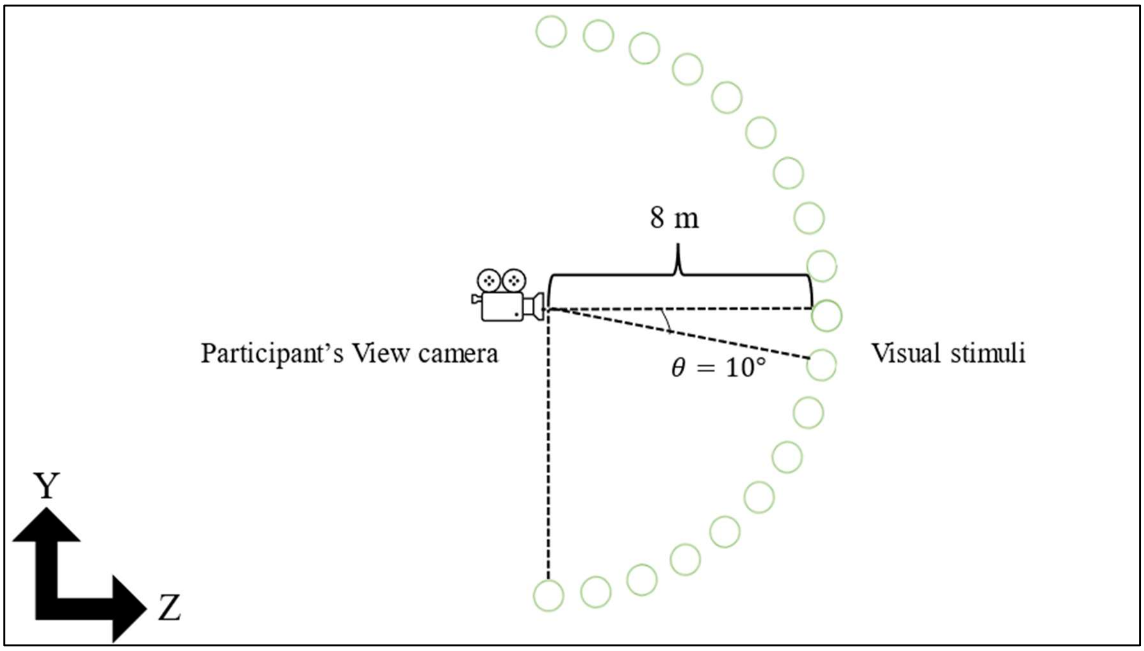


Figure 2-5-9 Experiment environment in VR space. (experiment 3)

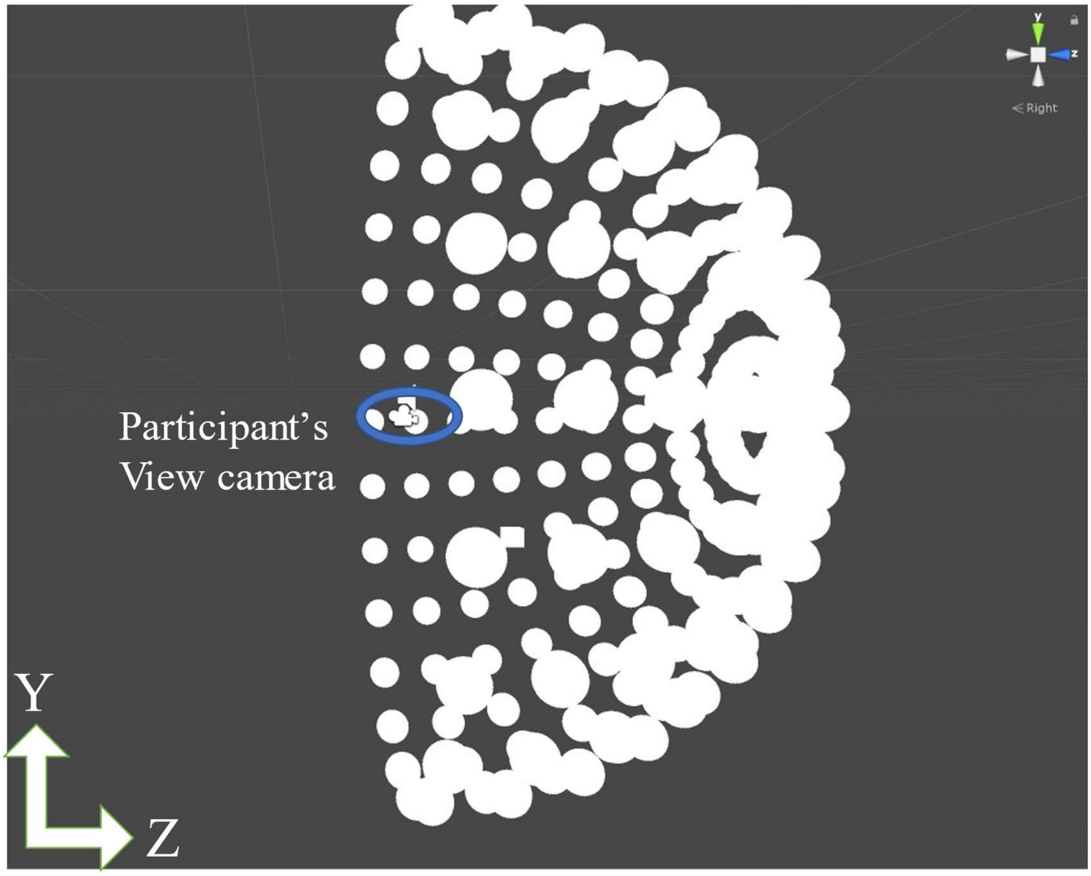


Figure 2-5-10 experiment environment (experiment 3)

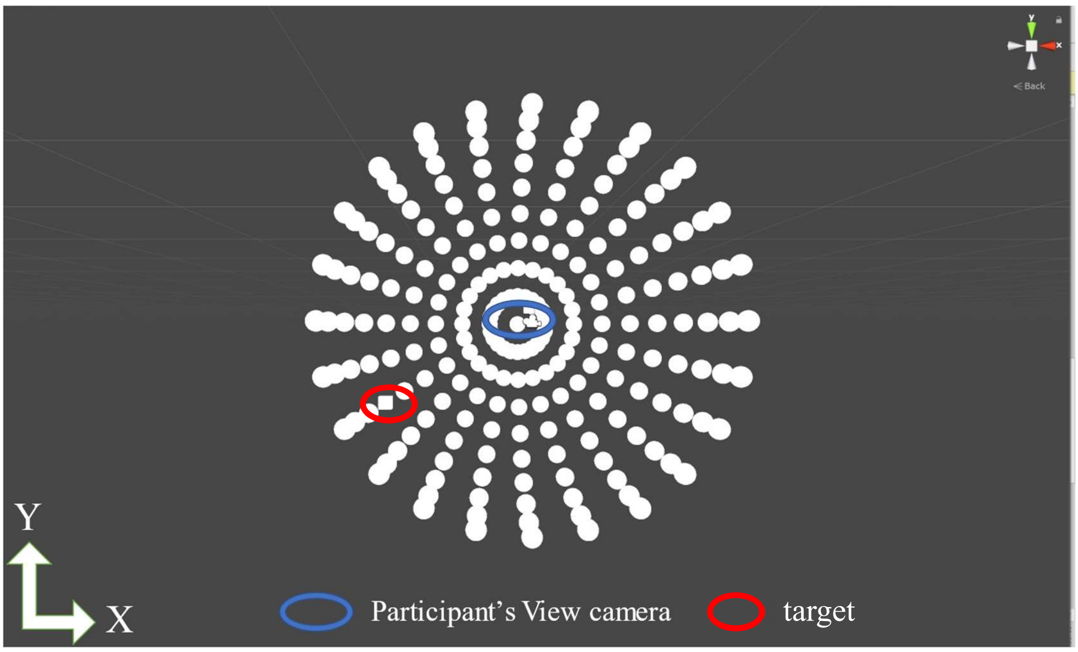


Figure 2-5-11 experiment environment (experiment 3)

Chapter 3 the Results and Analysis

3.1 Results

I. <u>Auditory-Visual Simon task (Experiment 1)</u>

The data obtained by the experiment were the reaction time for each trial (100 trials/person), the type of stimulus for each trial, and the correct and incorrect responses. From the data, we first excluded the wrong answers.

The acquired data were Z-transformed (standardized) for each individual to reduce the effects of individual differences and then aggregated to obtain a representative value for an individual. The reaction time after Z-transformation was plotted for each condition (figure. 3-1-1 ~ 3-1-4). The results are shown below. The result shows that Reaction time tended to be shorter when the spatial characteristics of the auditory stimulus (right or left) were congruent with the direction in which the response should be made (right or left). In addition, the reaction time was shorter when auditory and visual stimuli were presented simultaneously than when only auditory stimuli were presented. In particular. visual stimuli tended to shorten the reaction time when both auditory and visual stimuli were congruent with the direction to react. Visual_Stimuli is the variable that indicates the presence and direction of visual stimuli. Auditory_Stimuli is the variable that indicates the direction of auditory stimuli.

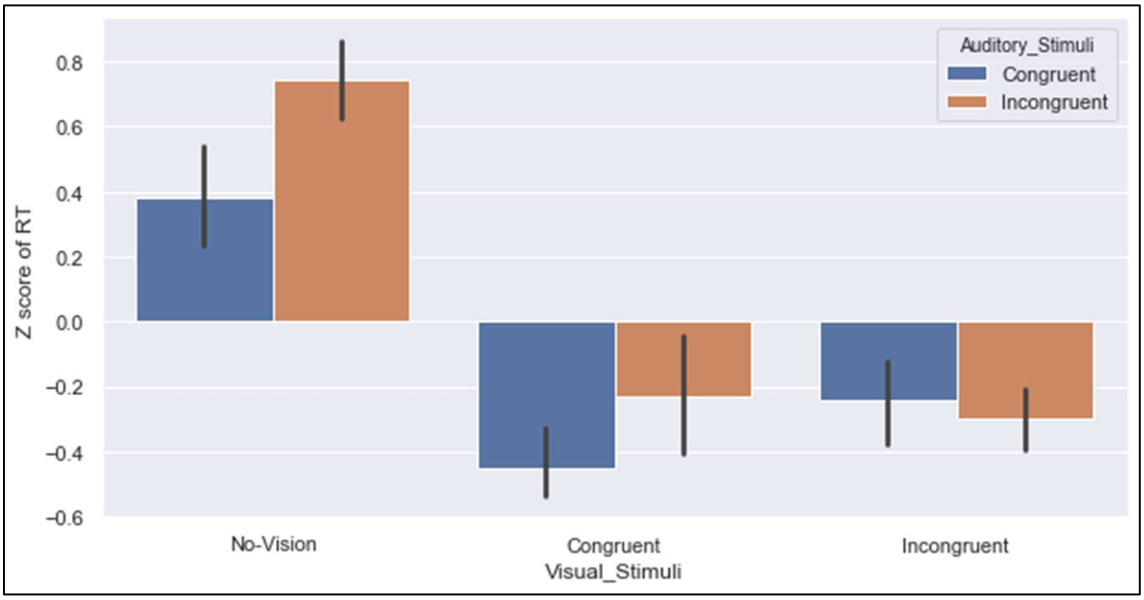


Figure 3-1-1 Z score of RT plotted by condition for visual and auditory stimuli. (Z-transformed and aggregated data)

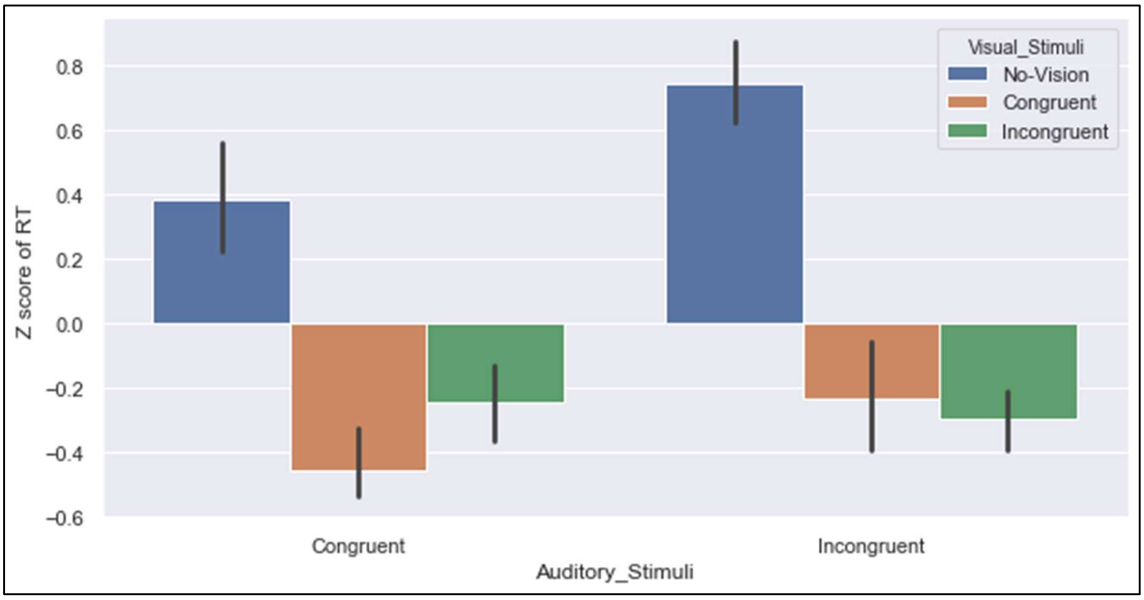


Figure 3-1-2 Z score of RT plotted by condition for visual and auditory stimuli. (Z-transformed and aggregated data)

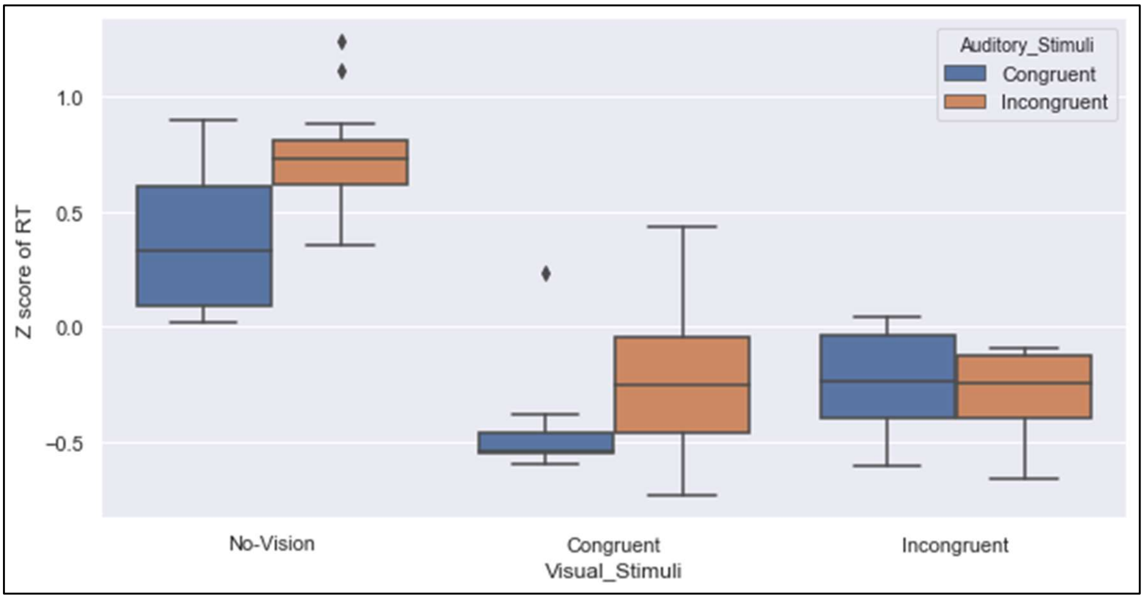


Figure 3-1-3 Z score of RT box-plotted by condition for visual and auditory stimuli.

(Z-transformed and aggregated data)

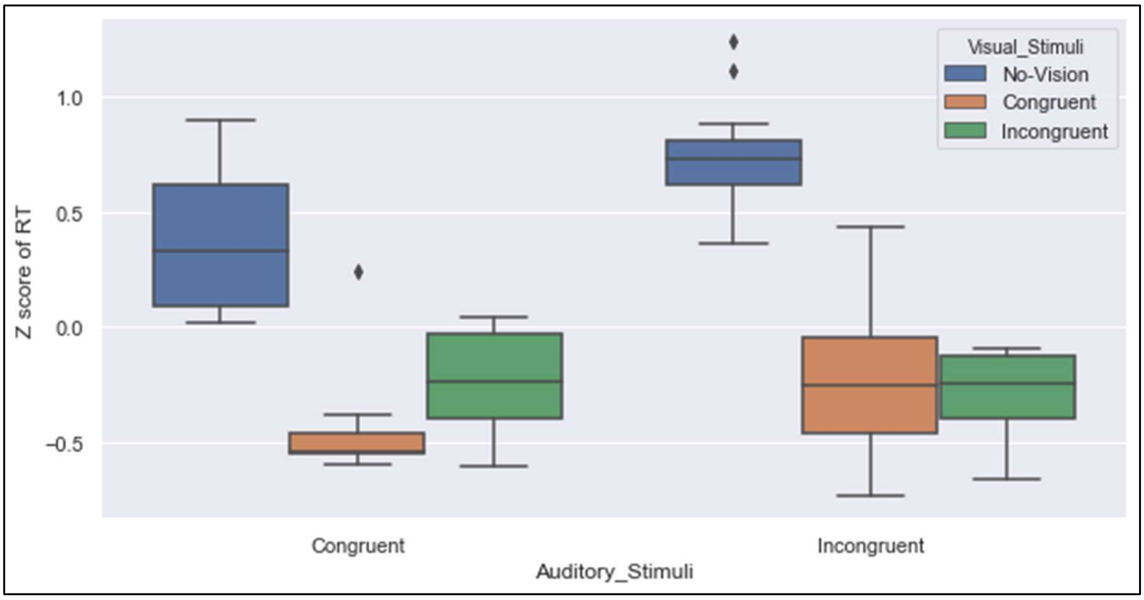


Figure 3-1-4 Z score of RT box-plotted by condition for visual and auditory stimuli.

(Z-transformed and aggregated data)

- Z-transformed Raw data analysis (Assume all data are independent)

The data are not guaranteed to be normal, but an approximation to the normal distribution by application of the Central Maximum Theorem is assumed. Two-way ANOVA was be conducted on the Z-transformed raw data, considering each of the multiple trials obtained from individuals as independent trials. Factor of Visual_Stimuli, factor of Auditory_Stimuli, and the interaction between Auditory_Stimuli and Visual_Stimuli were all statistically significant effect on RT($p<0.01$).

Table 1 Result of two-way ANOVA [4](Z-transformed Raw-data)

Two-way analysis of variance					
Source	SS	df	MS	F-ratio	p-value
Visual_Stimuli	217.5107	2	108.7553	135.0991	0.0000
Auditory_Stimuli	9.1695	1	9.1695	11.3907	0.0008
Auditory_Stimuli x Visual_Stimuli	10.1607	2	5.0804	6.3110	0.0019
Error	1034.4306	1285	0.8050		
Total	1273.6034	1290	0.9873		

[4] ISEKI's ANOVAKUN is used in R.

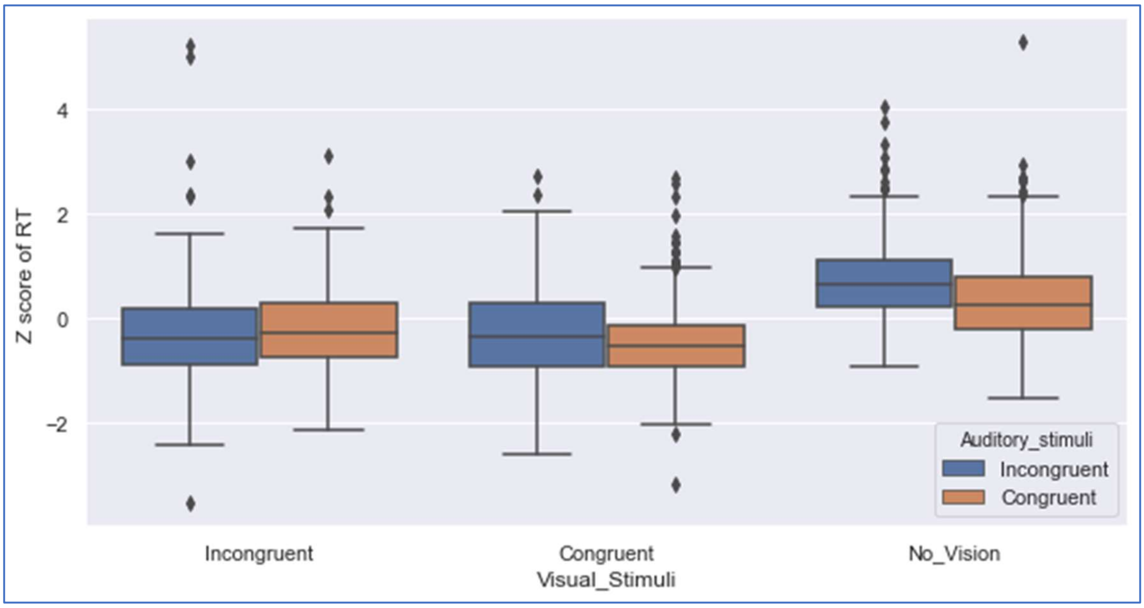


Figure 3-1-5 For reference: Z score of RT box-plotted by condition for visual and auditory stimuli. (Z-transformed raw data)

The distribution of the data is symmetrical compared to the post-aggregation one.

- Using Ex-Gaussian fitting (Assume all data are independent)

We did Z-transformation on each individual to analyze raw data rather than aggregated data and grouped the data according to the conditions of each stimulus. Then we fitted the raw data to ex-gaussian distribution to get the parameter μ, σ, τ. Both Auditory_Stimuli and Visual_Stimuli are Congruent (Group 3) had the shortest reaction time.

Table 2 Result of ex-gaussian fitting[5] of z-score RT. (Z-transformed Raw data).

Ex-Gaussian distribution fitting							
Condition Group		1	2	3	4	5	6
Condition	Auditory_Stimuli	Congruent	Incongruent	Congruent	Incongruent	Congruent	Incongruent
	Visual_Stimuli	No_Vision	No_Vision	Congruent	Congruent	Incongruent	Incongruent
Estimated Parameter	Mu(μ)	-0.3504	0.0901	-1.1092	-1	-0.7958	-0.9433
	Sigma(σ)	0.5344	0.5467	0.5378	0.6164	0.595	0.623
	Tau(τ)	0.7216	0.6391	0.6533	0.7552	0.5756	0.6577

5 "retimes" package is used in R.

- Aggregated data analysis (Consider within-participant factors)

This experiment was a within-subjects factorial design and was a repeated measurement with multiple trials under the same conditions for the same subject. Therefore, representative values were calculated and aggregated for each subject. In the case of the pre-aggregated data, the number of data was sufficient to use the Central Limit Theorem. In this case, however, the number of data was the same as the number of people in each condition. Therefore, it should be noted that the normality assumption for using ANOVA was not guaranteed. However, since there is no non-parametric test equivalent to two-way ANOVA, ANOVA was used.

The result shows that both factors of Auditory_Stimuli and Visual_Stimuli were significant effect on RT ($p<0.01$). Also, the interaction between Auditory_Stimuli and Visual_Stimuli was also statistically significant ($p<0.05$).

Table 3 Result of ANOVA[6] (Z-transformed and aggregated data)

Two-way analysis of variance					
Source	SS	df	MS	F-ratio	p-value
s	0.0782	12	0.0065		
Visual_Stimuli	13.191	2	6.5955	55.5902	0
s x Visual_Stimuli	2.8475	24	0.1186		
Auditory_Stimuli	0.598	1	0.598	13.6913	0.003
s x Auditory_Stimuli	0.5241	12	0.0437		
Visual_Stimuli x Auditory_Stimuli	0.577	2	0.2885	4.8319	0.0172
s x Visual_Stimuli x Auditory_Stimuli	1.433	24	0.0597		
Total	19.2489	77	0.2500		

[6] ISEKI's ANOVAKUN is used in R.

Multiple comparison was conducted to find out which specific groups were significantly different from each other. The results are shown below. The results showed that there was a significant difference in the reaction time between the congruent trials and incongruent trials for auditory stimuli without visual stimuli (Auditory Simon effect). The p-value was 0.0002441. On the other hand, the difference was decreased between the visual stimulus groups. In the vision-congruent group, P-value was 0.09424. And in the vision-incongruent group, P-value was 0.4973. These results indicated that the Simon effect was also present in auditory stimuli. Furthermore, it was found that the display of the visual stimuli reduced the difference in RT caused by auditory stimuli. The statistical hypothesis test revealed that there was a significant difference between the auditory stimulus congruent- incongruent in the absence of visual stimuli, while there was no significant difference between those in the presence of visual stimuli. As normality was not found when I checked that of RT, I used the Exact Wilcoxon signed-rank test, a non-parametric test. This result did not change if Holm method was applied. (corrected P-value)

Table 4. P-Value of Wilcoxon exact test for 2 groups in each visual condition. (Z-transformed and aggregated data)

Exact Wilcoxon signed-rank test			
Groups	Sound_Congruent-Sound_Incongruent Under No-Vision condition	Sound_Congruent-Sound_Incongruent Under Vision_Congruent condition	Sound_Congruent-Sound_Incongruent Under Vision_Incongruent condition
V-statistic	86	71	35
P-value	0.0002441	0.08032	0.4973
Corrected P-value	0.0007323	0.16064	---

- Using GLMM approach

The methods used so far have been analyzed after all kinds of processing on the data, such as Z-transform, aggregation, and not considering the distinction between subjects. Generalized Linear Mixed Model (GLMM)[XXXVII]was adopted for a more rigorous analysis of raw data. GLMM can model not only normal distributions but also any probability distributions, and it can handle data with multi-level structures generated by iterative trials. With this, the reaction time data of 13 participants could be analyzed separately for random effect and fixed effect. From previous studies, it is known that the reaction time is fitting to the gamma distribution.[XXXVIII] So, the distribution family was set to a gamma distribution. First, the intraclass correlation coefficients (ICC) were calculated to determine whether the experimental data should be subjected to multilevel analysis. First, we estimated the null model. Then, ICC is defined as Variance of intercept variability/ (Variance of intercept variability + variance of residue).[XXXIX] In this case ICC was 0.00226/ (0.02333+0.00226) = 0.088316 (table 5). Since this value is greater than 0.05, it was decided that GLMM should be used.

Table 5 Null model

Generalized linear mixed model fit by maximum likelihood. (Laplace Approximation)				
Family: Gamma (identity)				
Formula: time ~ (1 | ID) (null Model)				
Random effects:				
Groups	Name	Variance		Std. Dev
ID	(Intercept)	0.00226		0.04754
Residual		0.02333		0.15275
Number of obs.	1291	groups:	ID,13	

Next, we estimated the intercept variation model, which is the main issue. The estimation results of the model are shown below (Table 6, 7). These results also indicate that the factor of Auditory_Stimuli, factor of Visual_Stimuli, and their interaction terms were significant effect on RT, as were the results of the ANOVA. Auditory_Stimuli used Congruent as the baseline, and Visual_Stimuli used No_Vision as the baseline. It is worth noting that the reaction time was significantly shorter when both Auditory_Stimuli and Visual_Stimuli were Incongruent. (table 7)

Table 6 The intercept variation model

Generalized linear mixed model fit by maximum likelihood (Laplace Approximation)					
Family: Gamma (identity)					
Formula: RT ~ Auditory_Stimuli + Visual_Stimuli + (1 | ID)					
Random effects:					
Groups	Name	Variance		Std.Dev	
ID	(Intercept)	0.002105		0.04588	
Residual		0.019805		0.14073	
Number of obs	1291	groups:	ID,13		
Fixed effects:					
	Estimate	Std. Error	t-value	Pr(>|z|)	
(Intercept)	0.877609	0.045637	19.23	<2e-16	***
Auditory_Stimuli:Incongruent	0.014054	0.005686	2.472	0.0135	*
Visual_Stimuli: Congruent	-0.101922	0.007094	-14.367	<2e-16	***
Visual_Stimuli: Incongruent	-0.091393	0.007114	-12.848	<2e-16	***

Table 7 The intercept variation model (Interaction terms included)

Generalized linear mixed model fit by maximum likelihood (Laplace Approximation)					
Family: Gamma (identity)					
Formula: RT ~ Auditory_Stimuli + Visual_Stimuli + Auditory_Stimuli: Visual_Stimuli+(1 | ID)					
Random effects:					
Groups	Name	Variance		Std.Dev	
ID	(Intercept)	0.002108		0.04592	
Residual		0.019727		0.14045	
Number of obs	1291	groups:	ID,13		
Fixed effects:					
	Estimate	Std. Error	t-value	Pr(>|z|)	
(Intercept)	0.867893	0.04592	18.9	<2e-16	***
Auditory_Stimuli: Incongruent	0.033649	0.0104	3.235	0.00121	**
Visual_Stimuli: Congruent	-0.094188	0.009797	-9.614	<2e-16	***
Visual_Stimuli: Incongruent	-0.071083	0.010013	-7.099	1.26e-12	***
Auditory_Stimuli:Incongruent-Visual_Stimuli:Congruent	-0.015154	0.014161	-1.07	0.28456	
Auditory_Stimuli:Incongruent-Visual_Stimuli:Incongruent	-0.040728	0.014166	-2.875	0.00404	**

II. Audiovisual mental arithmetic task (Experiment 2)

The data obtained is the gaze data for each trial. First, we plot each participant's gaze data for each trial type as a vector-plot. Only during the auditory mental arithmetic task was there a change in gaze data. This change was observed in 4 of the 12 participants. We have included the eye plots of the remaining eight in the appendix. "Sound_addition (Sa)" is the eye data when an addition problem is auditorily presented, "Sound_subtraction (Ss)" is the eye data when a subtraction problem is auditorily presented, "Vision_addition (Va)" is the eye data when an addition problem is visually presented, and "Vision_subtraction (vs)" is the eye data when a subtraction problem is visually presented.

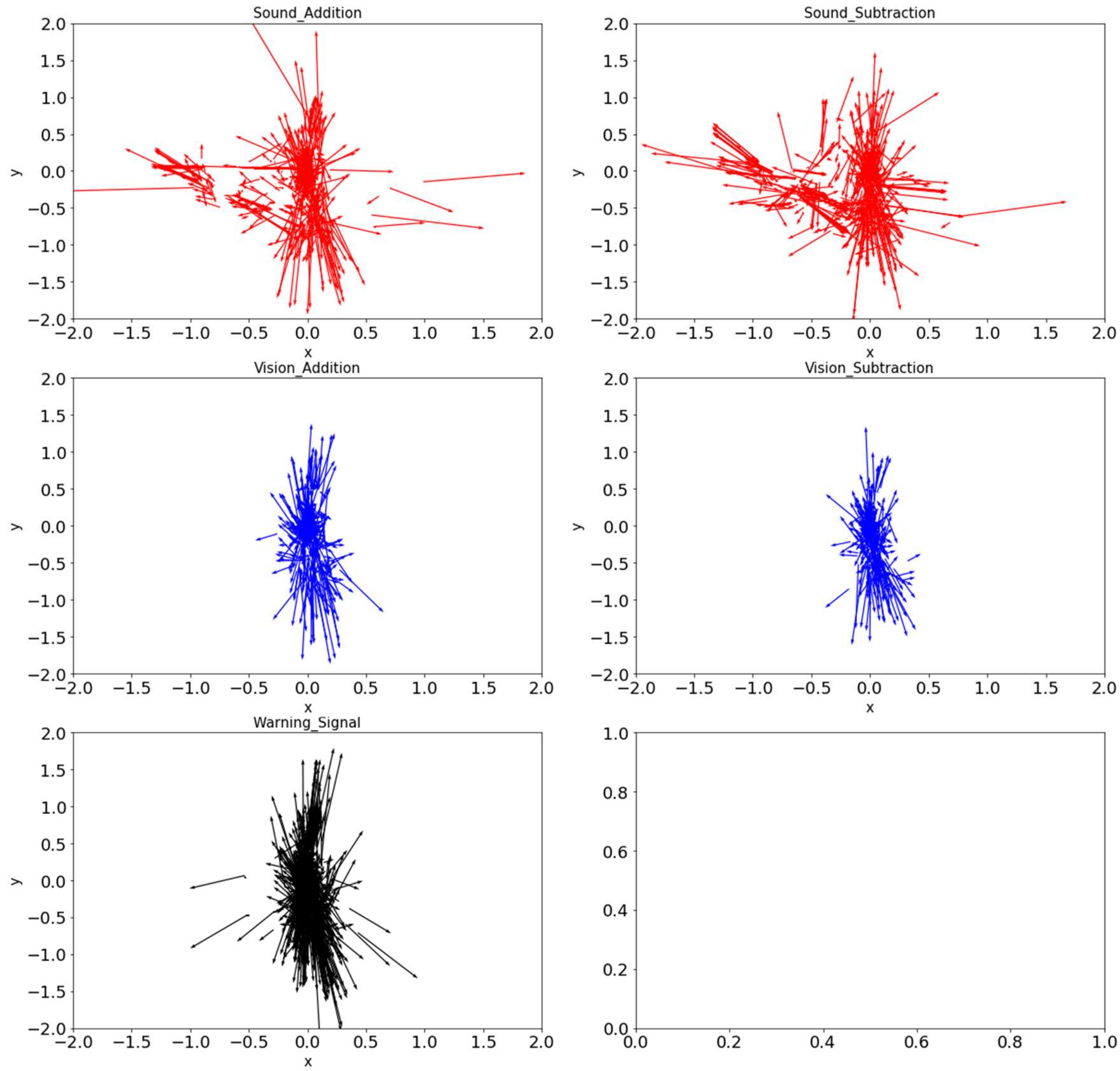


Figure 3-1-6 Vector Plot of Combined Gaze x, y. (ID2)

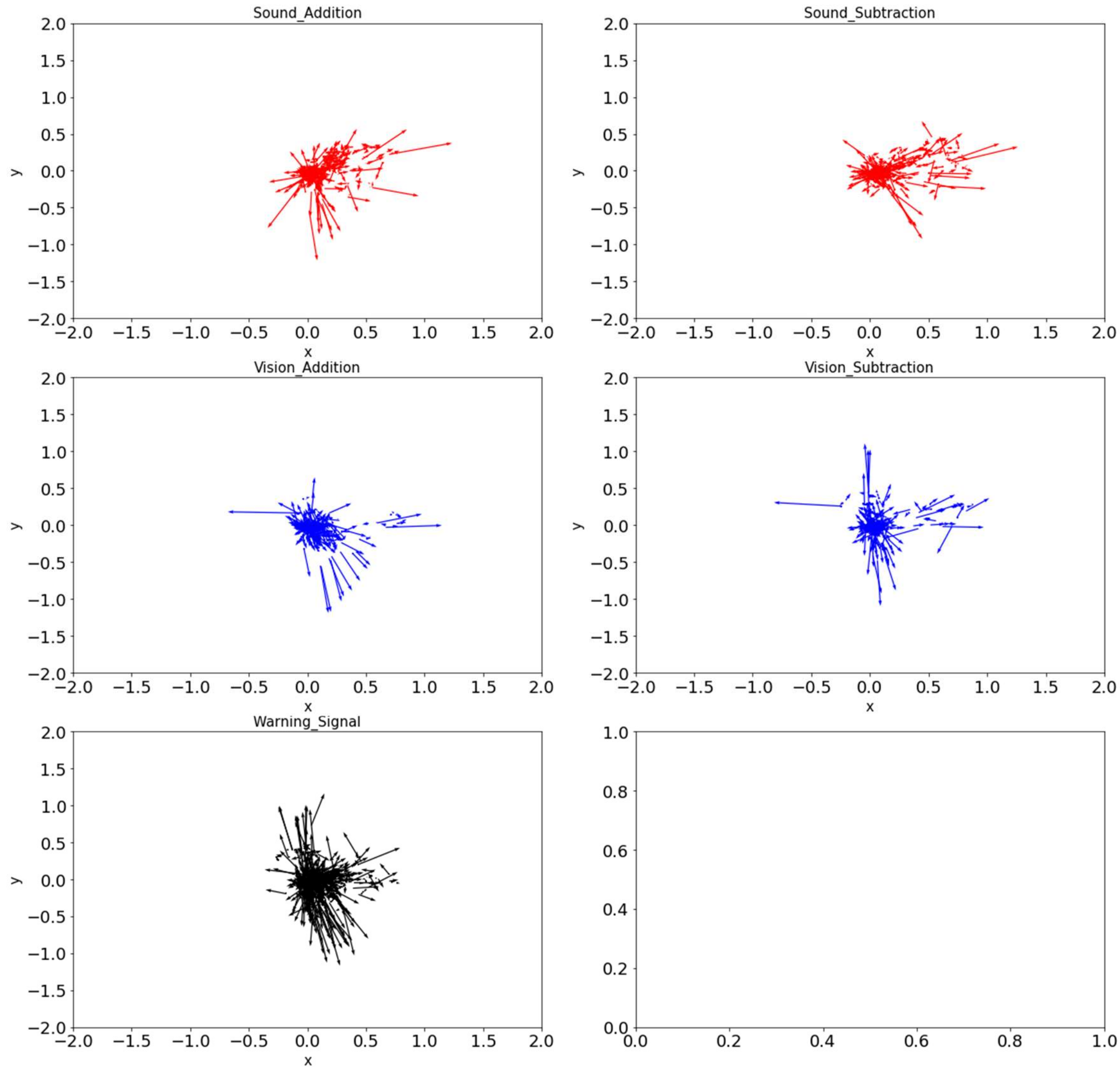


Figure 3-1-7 Vector Plot of Combined Gaze x, y. (ID3)

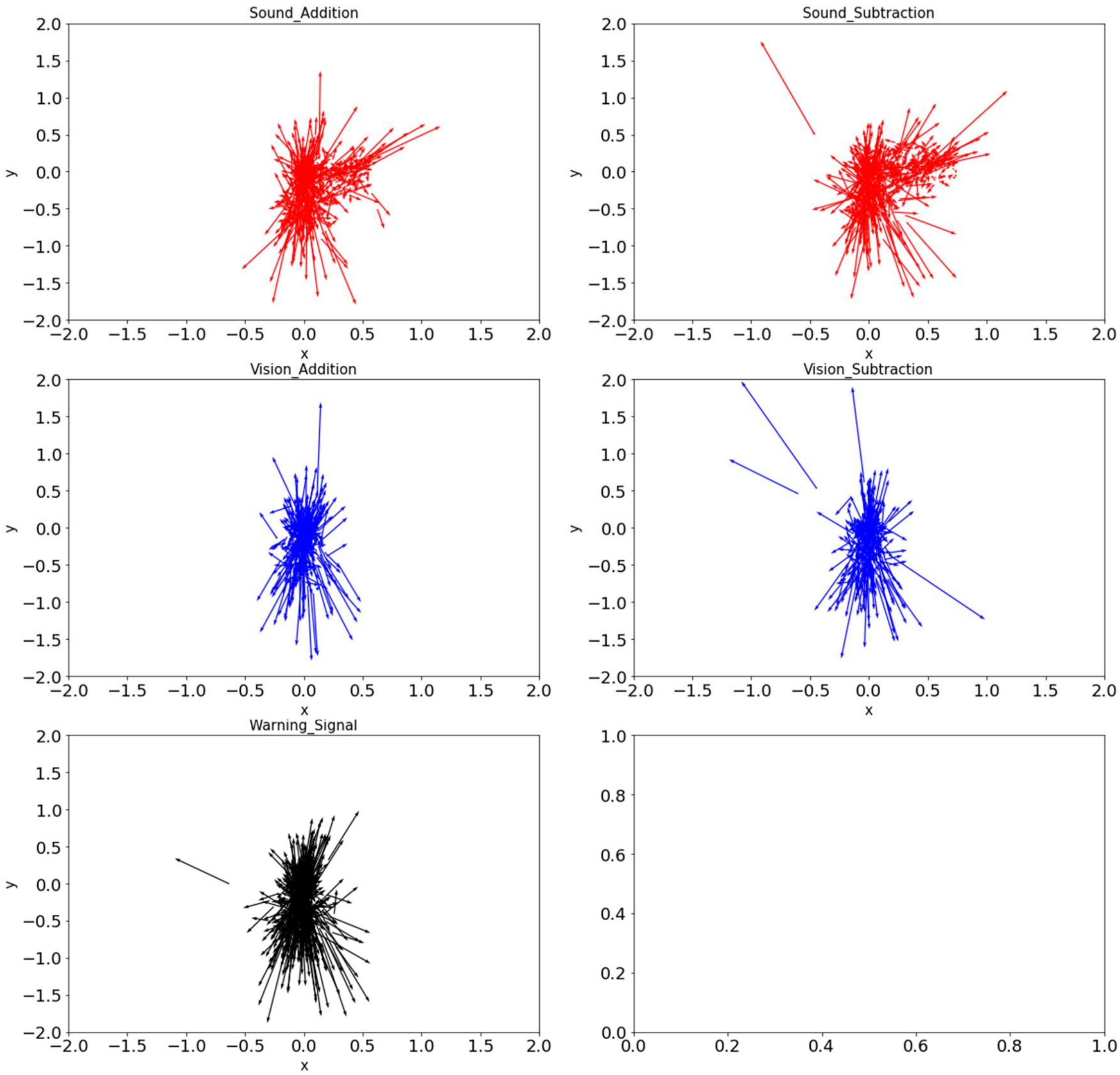


Figure 3-1-8 Vector Plot of Combined Gaze x, y.（ID5）

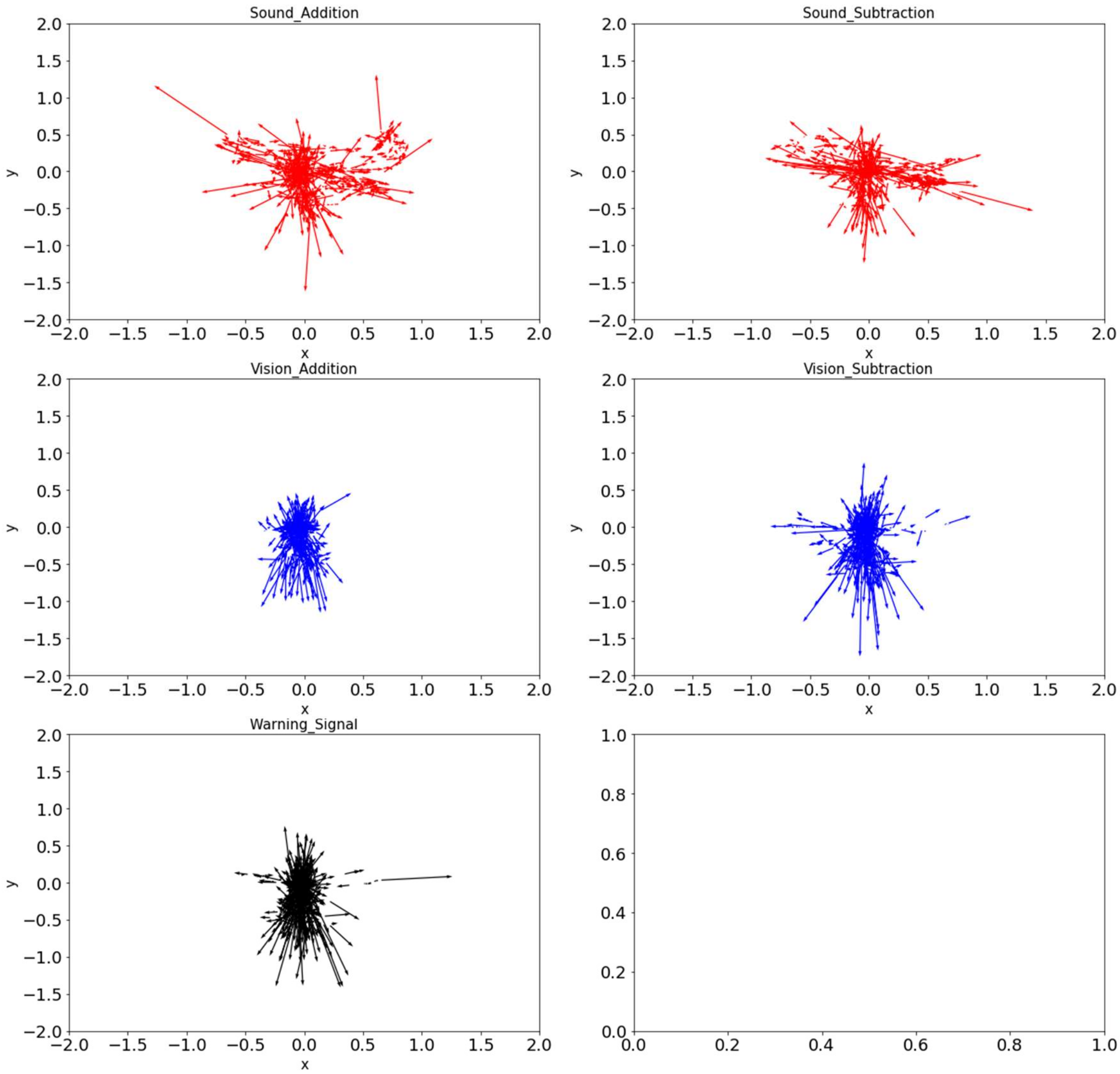


Figure 3-1-9 Vector Plot of Combined Gaze x, y（ID11）from the eye position.

To quantitatively evaluate the gaze data, we divided each subject's gaze data into type of trial and calculated the Shannon entropy.

The following is a definition of Shannon entropy based on references.[XL] Consider sample space $A = \{a_1, a_2, \ldots, a_n\}$ and $P(X = a_k) = p_k$. Then the Shannon entropy is represented by the following equation:

$$H(X) = -\sum_{k=1}^{n} p_k log p_k$$

This study used the previous research method.[XLI] The entropy was calculated and plotted as a box plot for each condition. The following table shows the entropy calculated by calculating the frequency distribution (bin=20) of the x- and y-coordinates of the eye data separately and the simultaneous entropy calculated based on the simultaneous probability of both the x- and y-coordinates. It was found that the entropy of gaze was greater when the subject was performing a mental arithmetic task compared to when a warning signal was presented.

For joint entropy, the Exact Wilcoxon signed-rank test was repeatedly used with the warning_signal group (control group) and the other four groups. The results were significant ($p<0.01$, V-statistics were Sa:66, Ss:66, and Va:63) except for the visual subtraction group($p>0.10$, V-statistic was 52). These results did not change with Holm's correction.

Table 8 Shannon entropy of gaze x-coordinate.

Shannon Entropy of gaze x [bit]

	Type of trial				
ID	Sound_addition	Sound_subtraction	Vision_addition	Vision_subtraction	Warning_signal
1	3.108736365	3.010054265	3.182950504	2.774932826	2.566261229
2	2.412718204	2.448988844	2.674121163	2.380321116	1.571043488
3	3.013465809	3.147283533	2.850298187	2.253421848	2.533905497
4	2.449808365	2.357920503	2.859958029	2.039607203	1.742129576
5	3.08424369	2.741809258	3.029257704	2.09809273	1.237263737
6	3.07636728	2.750378814	2.552865944	2.216436259	1.774531533
7	1.874494219	2.250888252	2.309794734	1.78752158	1.341271976
8	2.766497142	2.86075831	2.177274763	2.491060875	1.752597014
9	2.136421249	2.144828861	2.393527195	1.99426637	1.428173981
10	2.369954784	2.065402949	2.86663289	1.914527834	1.869548984
11	3.038552499	2.727082661	2.986997634	1.63331195	0.91496318
12	2.174850069	2.741507273	1.757703348	2.16089217	1.771585766

Table 9 Shannon entropy of gaze y-coordinate.

Shannon Entropy of gaze y [bit]

	Type of trial				
ID	Sound_addition	Sound_subtraction	Vision_addition	Vision_subtraction	Warning_signal
1	2.018138915	2.628543784	1.927412932	1.822786703	1.695508926
2	3.67520617	3.569854839	2.613573356	2.104992985	2.696898344
3	2.810866586	2.702464299	2.402284389	1.944646057	2.062403398
4	2.857299919	2.942594325	2.396326112	2.36482147	3.146786458
5	2.520013096	2.672387258	1.716875387	1.627740749	1.561851092
6	3.115125103	3.098636232	1.967344069	1.69249005	1.88511022
7	1.828327676	1.541703687	1.543618854	1.363912223	0.917854556
8	2.193702429	2.532962987	1.809590353	1.815882491	2.413148975
9	1.596040547	1.715126109	1.068998657	1.508291131	0.825408342
10	1.771146035	2.193602708	2.010301815	2.085326579	2.220527411
11	2.910284824	2.812524652	1.6995646	1.318038744	1.305374818
12	1.111827469	1.353614892	1.018815738	1.096535941	1.081623206

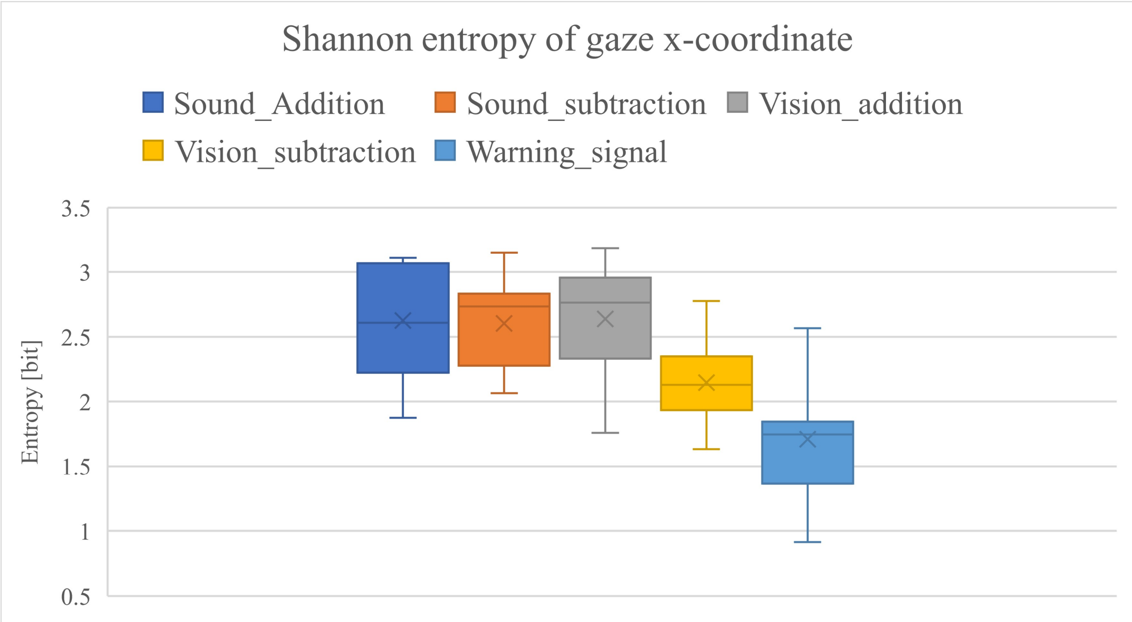


Figure 3-1-10 Bar plot of Shannon entropy of gaze x-coordinate.

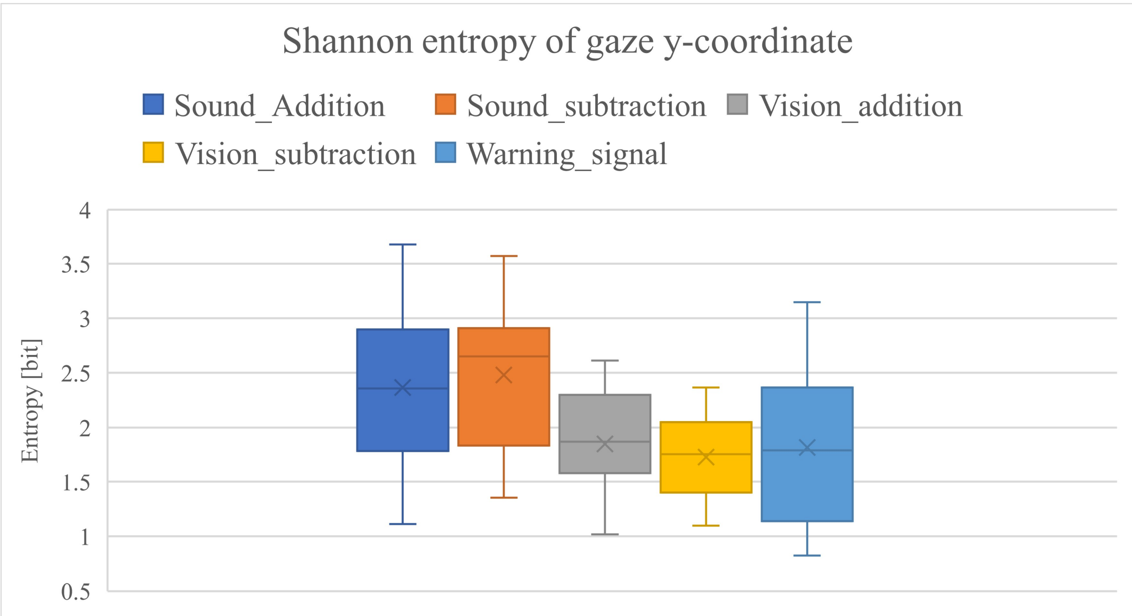


Figure 3-1-11 Bar plot of Shannon entropy of gaze y-coordinate.

Table 10 Joint entropy of gaze (x, y).

Joint Entropy of gaze (x, y) [bit]					
	Type of trial				
ID	Sound_addition	Sound_subtraction	Vision_addition	Vision_subtraction	Warning_signal
1	4.912283659	5.192900622	4.587756054	4.264536667	3.992398369
2	5.307475779	5.325742036	5.007473185	4.204486246	4.05211146
3	5.157894585	5.048946656	4.636915236	3.900036218	4.247542903
4	4.937735133	4.688938678	5.009851273	4.124478447	4.574819
5	5.167458534	4.851492646	4.590687122	3.540476132	2.616997822
6	5.686496129	5.263451355	4.241076814	3.603556854	3.317472663
7	3.296385311	3.404789715	3.58792005	2.858326693	1.975049313
8	4.540463636	4.824350974	3.728498055	4.049403656	3.869022044
9	3.392438004	3.54441909	3.306350461	3.252870969	2.110929323
10	3.739022213	3.840223786	4.562894475	3.699873988	3.718292938
11	4.696248008	4.59009412	4.488310703	2.767438174	2.076406049
12	3.054646343	3.997457608	2.663958517	3.006186307	2.749154534

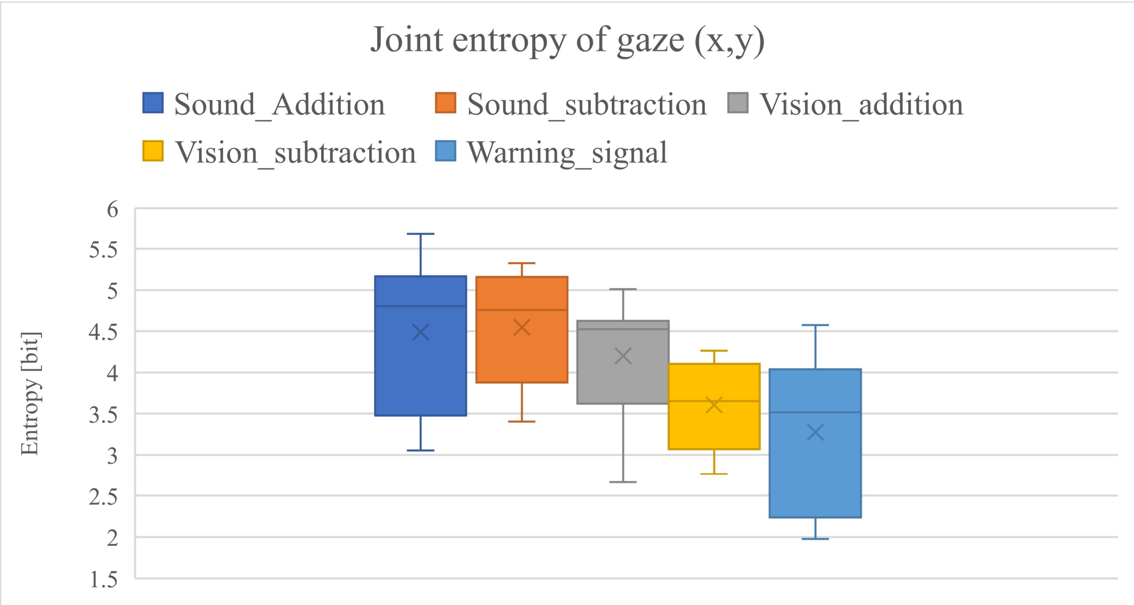


Figure 3-1-12 Bar plot of Joint entropy of gaze (x, y).

III. Experiment 3

The data obtained are the angular velocity and acceleration of the head and the position and rotation angle. The analysis was done for each block. In the first trial of the subtraction block, the sound of addition was mixed in. In addition, there was a lag in the transition of blocks. For this reason, we excluded the first trial of each block and analyzed the other 71 trials x 3 blocks. To see if there was a change in the way the head moved for each part, we calculated the mean of the angular velocity for each part. Table11 is the average of the angular velocity.

Table 11 angular velocity of participant's head during experiment.

The mean value of Angular velocity [rad/s]			
	block		
ID	No_Sound	Addition	Subtraction
1	0.76926018	0.86080252	0.91817622
2	0.91807326	0.92872425	1.10314654
3	1.14058367	1.17712493	1.22367524
4	1.2411803	1.46879751	1.30196154
5	1.68963284	1.36639557	1.52444177
6	0.91115578	0.90179271	0.9487747
7	1.45753166	1.52700805	1.50625005
8	2.60125454	2.62698159	2.49057987
9	1.56947612	1.66211306	1.59908025
10	1.67955077	1.77116344	1.90872209
11	1.08779985	1.12449212	0.96128115
12	1.16058917	1.07763941	1.10473331
13	1.6980298	1.67419887	1.63138677
14	0.69108915	0.82387184	0.82068718
15	1.72224547	1.87399233	1.84569023
16	1.25154153	1.55749352	1.34961519
17	1.52933558	1.70071739	1.53571609
18	0.89156618	1.08366174	0.89594463
19	1.27877452	1.36840974	1.33798984
20	1.16468106	1.17203762	1.17853488
21	1.23587617	1.66011826	1.73059947
22	1.2471283	1.25551243	1.31134665
23	1.1279609	1.31996807	1.22244112
24	1.47393492	1.3932158	1.46716532
mean	1.314094	1.390676	1.371581

Then, determined by paired samples t-test if there is a difference in the mean of the angular velocity per block. The t-test is the test if there is a difference in the mean of the two groups. In this case, we compare the visual search performance when the arithmetic task is not imposed with that when arithmetic task is imposed. Therefore, we adopted the paired-samples t-test.

To quantitatively evaluate the differences, we conducted Dunnett contrasts were used as a multiple comparison method. The significance level is set at 0.05.

This result shows a significant difference in the absolute value of head angular velocity between the No-sound block and the Addition block(p=0.0119). On the other hand, there was no significant difference between the No-sound block and the Subtraction block(p=0.0672).

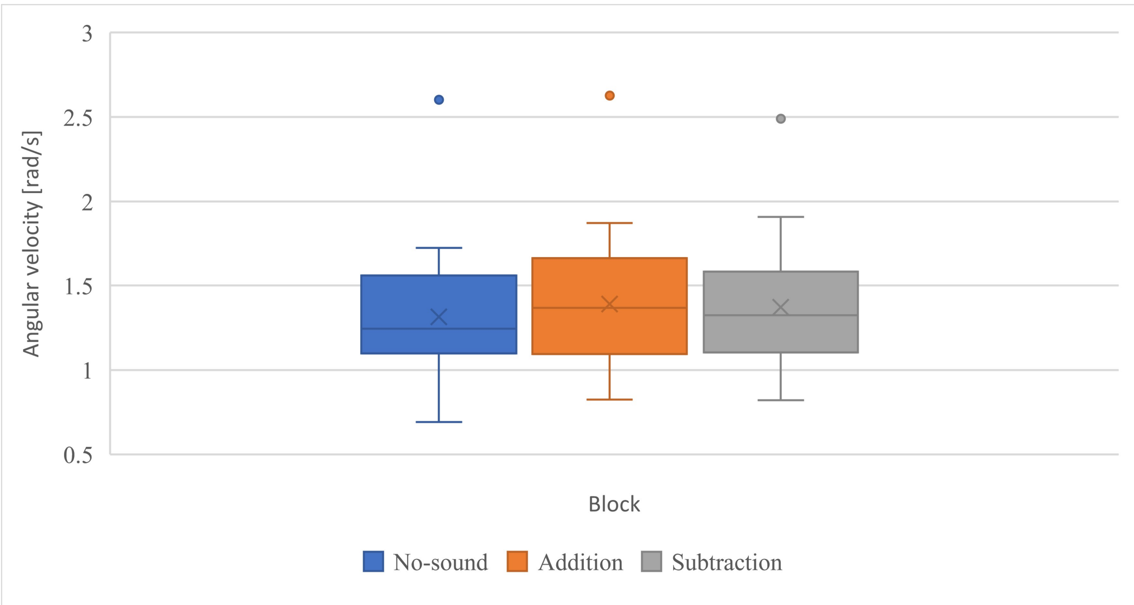


Figure 3-1-13 Bar-plot of the mean value of angular velocity. (n=24)

Table 12　Comparison of angular velocity Means using Dunnett contrasts.

Multiple Comparisons of Means: Dunnett Contrasts

	Estimate	Std. Error	t-value	Pr(>|t|)
Subtraction -No_sound	0.05749	0.02674	2.15	0.0672
Addition - No_sound	0.07658	0.02674	2.864	0.0119

For each part, the mean of Reaction time (search time) was calculated by individuals. Here is a boxplot of aggregated RT. In the mental arithmetic block, reaction time was shorter than in the No_sound block. For statistical evaluation, Wilcoxon signed-rank test was used for aggregated RT. In consideration of multiple comparisons, Holm correction was used.

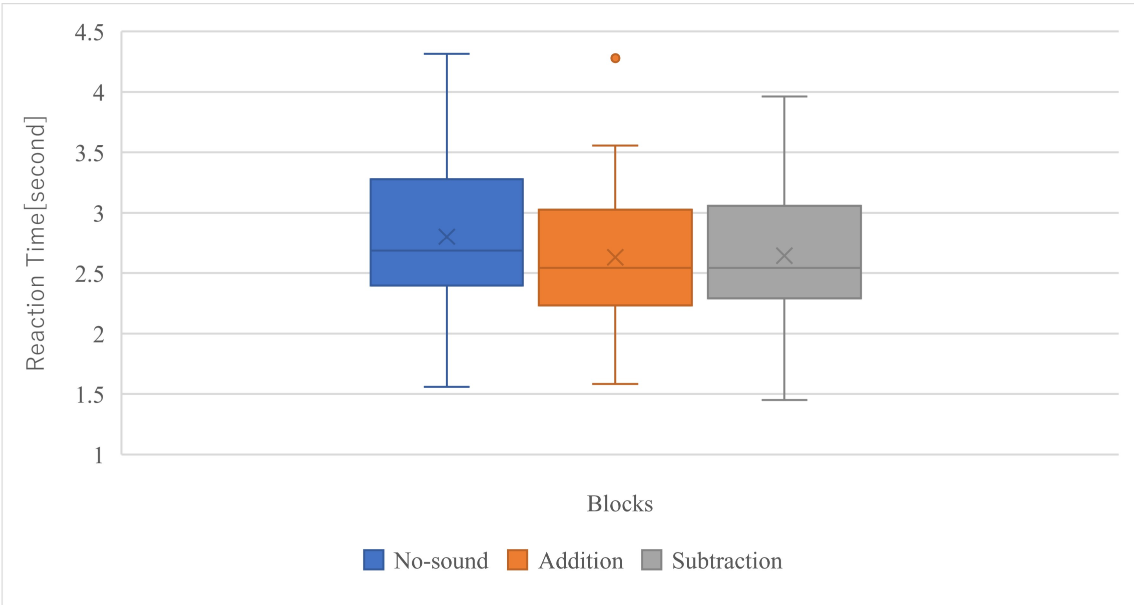


Figure 3-1-14 boxplot of the mean value of RT (n=24)

Table 13 Result of exact Wilcoxon signed-rank test of RT's mean (n=24).

Exact Wilcoxon signed rank test			
Groups	No_sound-Addition	No_sound-Subtraction	Addition-Subtraction
V-statistic	233	234	162
P-value	0.01641	0.01504	0.07469
Corrected P-value	0.03282	0.04512	0.07469

3.2 Statistical methods

This study uses many statistical methods. This section provides an overview of basic statistical knowledge and methods. The field of statistics used in this research is called inferential statistics. This field is mainly used to infer the characteristics of a population from the results of a sample survey for an event that cannot be surveyed in its entirety. Here described procedure of the statistical test. First, formulate the null hypothesis, alternative hypothesis, and the significance level. Then calculate the probability density of the null hypothesis. Finally, estimating how likely the actual outcome is to occur in the probability distribution. If the probability is over the significance level, the null hypothesis is adopted. On the other hand, if the probability is under the significant level, the alternative hypothesis is adopted.[XLII]

Follow describes these two statistical tests based on reference.

Two samples t-test is used for independent two groups to test the differences in the population mean. Consider sample A whose size is m extracted from a normal population of mean μ1, and variance σ1 and sample B whose size is n extracted from a normal population of mean μ2 and variance σ2. The null hypothesis here is "no difference in the mean of the two groups." $\bar{A}$ is the sample mean of A. $\bar{B}$ is the sample mean of B.

$$H_0: \mu_1 = \mu_2$$

When the variances of the two groups are equal, the estimate of the variance is:

$$s^2 = \frac{(m-1)s_1^2 + (n-1)s_2^2}{m+n-2}$$

s_1^2, s_2^2 are sample variance of each sample. Then, t-statistics is:

$$t = \frac{\bar{A} - \bar{B}}{s\sqrt{\frac{1}{m} - \frac{1}{n}}}$$

when variance s_1^2, s_2^2 are not the same, t-statistic is:

$$t = -\frac{\bar{A} - \bar{B}}{\sqrt{\frac{s_1^2}{m} - \frac{s_2^2}{n}}}$$

Whether or not to reject the null hypothesis is determined by whether or not these t-static is in the rejection region.[XLIII]

Paired t-test uses the difference of the two paired groups to find out there is the difference in the mean between two groups. $\overline{X_d}$ is the mean of the difference, n is the sample size, and S_d^2 is the sample variance. It then produces the following t-statistic:

$$t = \frac{\overline{X_d} - \mu_d}{S_d/\sqrt{n}}$$

The null hypothesis here is "no difference in the population mean of the paired groups". It means that:

$$H_0 : \mu_d = 0$$

$$t = \frac{\overline{X_d}}{S_d/\sqrt{n}}$$

Then, compare the t-statistic obtained here to the t distribution table. We can reject the null hypothesis if it falls below the percentage point corresponding to significance level.[XLIV]

3.2.1 Reaction time analysis

The objective variable of the data handled in this study is Reaction Time (RT). RT is a feature often used in psychological research. There are two types of reaction time: simple reaction time (CRT), in which participants only press the one button when they perceive a stimulus, and selective reaction time (SRT), in which they judge whether to press the button depending on the content of the stimulus.[XLV] This study mainly used the choice reaction time. The method of analyzing RT needs to be considered because RT data differs from other data. RT data does not follow a normal distribution. Parametric tests such as t-test and ANOVA are based on the assumption that the statistical population of the data follows a normal distribution. Therefore, it is not desirable to apply the method to row RT data. If applying, the possibility of Type 1 error increases.[XLVI] In addition, using representative values such as the mean of RT to make comparisons between groups is problematic. Therefore, there are three types of measures to avoid problems used in the RT study. In this chapter, the method used in this study is described based on Iseki and other's papers and books.

I. Z-conversion

The Z-transformation is a conversion operation that subtracts the mean value from each raw reaction time data and divides it by the standard deviation. This operation eliminates the individual differences in reaction time among subjects.[XLVII]

II. Fitting to ex-gaussian distribution

As described above, reaction time does not follow a normal distribution. An ex-gaussian, which is a hybrid of an exponential distribution and a normal distribution, is known as a distribution that often follows RT data distribution. By fitting RT data to this distribution, parameters such as mu, sigma, and tau are obtained. These tell us the shape of the distribution. Each parameter has a different meaning. mu has a meaning similar to that of the mean in the normal distribution of reaction times, and sigma represents the variation in reaction times, tau represents the positive skewness of the reaction time distribution.[XLVIII][XLIX] This enables reaction time analysis that takes into account not only the mean value but also the skewness of the distribution.

III. Generalized Linear Mixed Model and Non-parametric test.

Conventional statistical analysis methods have assumed that the population distribution is a normal distribution. At present, there are various methods to name a few, non-parametric test, the Generalized Linear Model (GLM), decision tree analysis. The following explanation about the Generalized Linear Mixed Model is described. In this experiment, the experiment was conducted multiple times on a single subject (figure 3-2-1-1). That is why multiple reaction time data is obtained. Also, the objective variable is reaction time and, therefore, does not follow a normal distribution. The GLMM model solves these problems. GLMM (Generalized Linear Mixed Model) is composed of two effects, which are Random effect and Fixed effect. Random effect treats individual differences. This allows to include not only traditional fixed effects, but also variant effects such as individual differences and trial differences of participants in the model.[L] GLMMs can also assume probability distributions other than the normal distribution, such as the gamma distribution. Furthermore, it is suitable for the analysis of experiments where there are multiple data for the same person and same conditions with repeated measurements, such as Experiment 1 (fig.3-2-1-1).

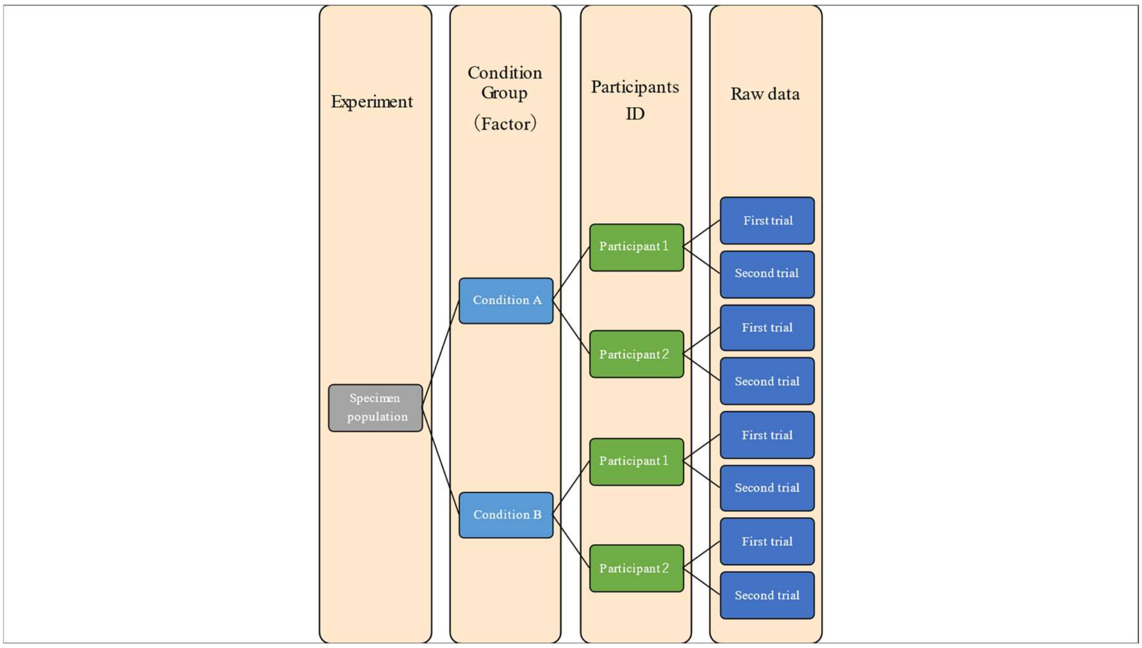


Figure 3-2-1-1 Diagram of multi-level structure in experiment 1 prepared by reference to the description and figure in 反応時間ハンドブック[LI] p96.

Also, this study uses a non-parametric test. Here are two of the most common methods. The first one is U-test. This is used to analyze the difference between the two groups. The U-test can be used for data whose population normality is not guaranteed but whose equivariance is guaranteed. It tests whether there is a difference between two independent sample populations. The number of samples of the two groups is placed as n1 and n2, respectively, and the two groups are mixed and ranked in decreasing order, and the sum of the two groups is the test statistic U.[LII]

If the number of n_1 and n_2 is large enough, $\mu_U = \frac{n_1 n_2}{2}$, $\sigma_U = \sqrt{\frac{n_1 n_2 (n_1 + n_2 + 1)}{12}}$.

Then, Z is the following equation:

$$Z = \frac{U - \mu_U}{\sigma_U}$$

Z can be considered to follow a normal distribution, and this can be tested.[LIII] The specific theory of hypothesis testing is described in the explanation of the t-test in 3.2. This is a non-parametric unpaired sample test. There is also a non-parametric paired samples test. It is called Wilcoxon signed-rank test. In this test, the differences between the paired data are determined and ranked in order of increasing absolute value. And, then Statistics V is calculated. V is defined as the sum of the rank whose sign is +.[LIV] In R, some functions may have slightly different definitions.

3.2.2 About multiplicity issue

This study often uses statistical tests such as the T-test and the Wilcoxon signed-rank test. Repeatedly using these tests can cause increasing the probability of Type I errors. Let us show two examples. First of all, there is a problem with multi-group comparison. When the mean of the three groups is compared, the t-test must be done three times. The null hypothesis is $\mu_1 = \mu_2 = \mu_3$, Only once every three times the result of the t-test, which is significant, can lead to rejecting the null hypothesis.[LV] If the null hypothesis is true, the probability of adopting the null hypothesis is $(1 - 0.05)^3 * 100 = 85.7\%$. It means the overall significance level is 14.3%. This result is over the significance level of each test (5%). Hence, there are several tools to fix it. Here are three typical examples. The first is the Bonferroni correction. This method is that each test's significance level is defined as the overall significance level divided by the number of times the test is done. (e.g., in the case that the overall significance level is 0.05 and the test is done three times, the significance level of each test is 0.05/3)[LVI] By adjusting the significance level of individual tests, the overall significance level can be set to any desired level. This method can easily apply to both parametric tests and non-parametric tests. Nevertheless, there is a problem with this method. It is so conservative that it can correct the probability of type1 error, but the probability of type2 error can be increased, which leads to a decrease in statistical power.[LVII] Recently, with the development of computers, improved multiple comparison methods have been published. The other method is Dunnett's test. It is multiple comparisons with the control group only, which is often used for medical fields.[LVIII] It can be used not only unpaired-samples but also paired-samples.[LIX] Holm correction is an improved version of the Bonferroni correction. In Holm correction, the p-values to be used for the multiple comparison test are ranked in decreasing order. For the smallest p-value, the significance level α divided by the number of tests (N) (α/N) is set as the first rank significance level, and if the null hypothesis is rejected, the second rank significance level is set as α/(N-2) to examine the significance for the second smallest p-value. Similarly, the third rank significance level is set to α/(N-3). If the null hypothesis is not rejected, the significance of the rank below it is withheld.[LX] In this study, multiple comparisons were made by multiplying the obtained p-values by N, (N-1), and (N-2) rather than adjusting the significance level.

3.2.3 Eye-tracking analysis

In this study, Eye-tracking data is obtained with FOVE0. During the experiment, the position and direction of the left and right eyes are recorded separately. However, in order to make the analysis easier, we used the Combined Gaze function, which is available from FOVE. It is the function that returns combined the vector of the right and left eyes. What is more, FOVE checks the reliability of the left and right eye data and returns the most reliable eye data.[LXI]

3.2.4 Head-tracking analysis

Head tracking data this study treated is the position of the head and the Euler angle. Euler angles are an angular system commonly used in Unity but let us do a supplemental explanation because it is hard to understand intuitively. There are several ways to describe the postural angle of the head. Typical ones are described below: pitch, yaw, low angle, and Euler angles.

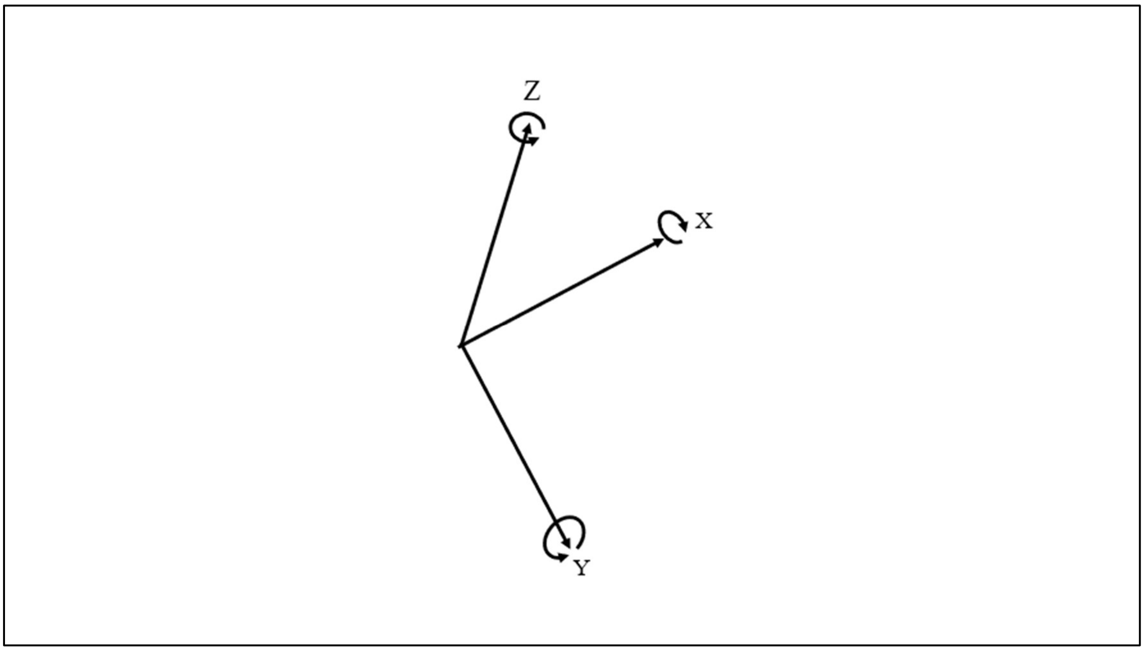


Figure 3-2-4-1 pitch, yow, roll angles.

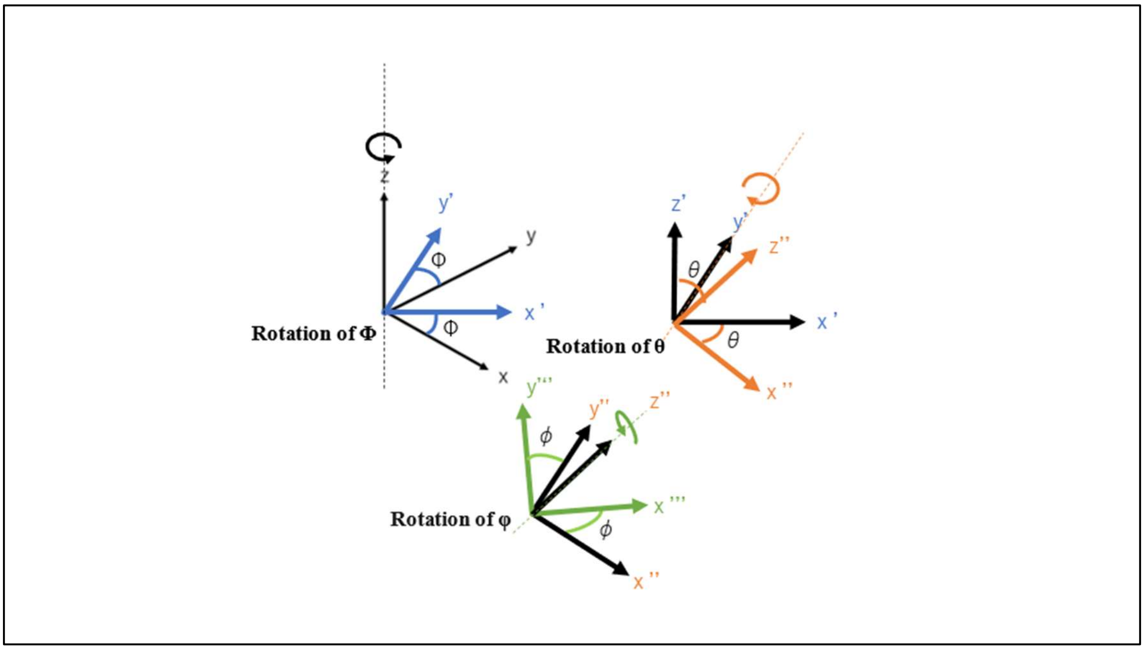


Figure 3-2-4-2 Euler angles.

These rotations can be shown as a matrix. The following is a matrix representation of the Euler angles based on reference.[LXII]

$$R(\varphi) = \begin{pmatrix} cos\varphi & sin\varphi & 0 \\ -sin\varphi & cos\varphi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$R(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & cos\theta & sin\theta \\ 0 & -sin\theta & cos\theta \end{pmatrix}$$

$$R(\Psi) = \begin{pmatrix} cos\Psi & sin\Psi & 0 \\ -sin\Psi & cos\Psi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

In this experiment the data is recorded every 0.02 seconds. The amount of data in each block is sufficiently large. Therefore, central limit theorem can be applied. the mean of the angular velocity approaches the population mean and follows a normal distribution.[LXIII] Therefore, parametric tests can be used. Typical tests are analysis of variance (ANOVA) and t-test.

Chapter 4 Discussion

4.1 Experiment 1

Results suggest the existence of the Simon effect by auditory stimuli and the existence of effect caused by visual stimuli. Also, regardless of the visual stimulus's spatial location, the presentation of the visual stimulus shortened the reaction time. The redundancy signal effect can explain this. Moreover, it is noteworthy that the graph suggests that when one modality stimulus is inconsistent with the spatial characteristics of the response, the Simon effect of the other modality stimulus seems to be reduced or eliminated (see Table 4, Figure 3-1-1, 3-1-3. When the visual stimulus was not presented, the auditory Simon effect existed. However, when the visual stimuli were presented, there was no significant difference in RT caused by the auditory Simon effect.) It is possible that the Simon effect caused by the spatial location of auditory stimuli was weakened because humans give priority to information obtained from vision. Alternatively, in multimodal perception, in addition to the SRC effect caused by the spatial location between stimulus and response, there may be an effect caused by the congruency of the spatial locations of the two different modalities stimuli. In fact, the GLMM results suggest the existence of an interaction term that significantly shortens reaction time when both Visual_Stimuli and Auditory_Stimuli are Incongruent. This result is curious. This is because the reaction time was shortened even though the visual and auditory stimuli were presented in the opposite direction to which they should have responded. This result can also be seen from the reaction time boxplot.

In addition, the fact that the reaction time changed with the presentation of two different modality stimuli and that the interaction term between the visual and auditory stimulus factors was significant in ANOVA and GLMM supports the coactivation model for multimodal perception of visual and auditory sensations.

4.2 Experiment 2

Vector plot results show that human tends to move eyes when calculating mentally Auditorily presented arithmetic tasks. Four people out of twelve showed this tendency. In particular, only when doing the auditorily presented arithmetic task, horizontal eye gaze movement's change was observed. This experiment only set a visual stimulus at the position (0,0,9.57). Nevertheless, the distribution of the gaze from left to right can be attributed to auditory stimuli of calculation problems and doing mental arithmetic. This suggests that auditory perception may be influencing the visual function of gaze. Therefore, this also indicates that auditorily presented arithmetic tasks can cause spatial attention shifts. There was a difference in plot between modalities, but not between addition and subtraction. The previous study reported that there is a difference in eye movement during auditorily presented mental arithmetic tasks between Addition and Subtraction. Nevertheless, this experiment did not observe it. The difference in experiment procedure may be one cause.

Besides, the Shannon entropy of the gaze data in the x-coordinate increased when doing visually or auditorily presented arithmetic tasks compared to when seeing the warning signal presentation, and the entropy in the y-coordinate also tended to increase when doing the auditorily shown task. This result is consistent with previous studies showing that number processing influences spatial attention. Therefore, it also supports that the mental arithmetic task, especially the auditorily presented arithmetic task, has some effect on visual function.

4.3 Experiment 3

The most interesting result of this experiment is that the visual search task's reaction time was shorter under multi-tasking. If attentional resources are identical for visual and auditory stimuli, attention is divided between the two tasks, and performance is expected to decrease. However, statistical tests show that reaction time in multi-tasking was significantly shorter than reaction time without sound. This result can lead to two hypotheses. The First thing is that the visual search task and auditory mental arithmetic task use different attention resources. This means that there is no conflict between unrelated tasks between visual and auditory. On the other hand, facilitating reaction time by multi-task may owe accessory effect or redundancy signal effect. The visual search's reaction time and the mean angular velocity of the head changed significantly under multi-tasking. The result of Experiment 2 suggests Auditory mental arithmetic tasks can cause spatial attention shift. The result of experiment 3 also supports this suggestion. We did not provide evaluation items such as the percentage of correct answers. Thus, we were unable to evaluate erroneous operations due to multi-tasking in different modalities. However, the fact that the reaction time under multi-tasking was shortened in the visual search task suggests that auditory perception affects the function of visual attention in humans.

Chapter 5 Conclusion

This study accomplished three things. First of all, we were able to set a precedent of cognitive research using VR space and head/eye-tracking. More and more VR will be a standard tool for our society. This study used the latest technologies, such as stand-alone VR and Unity, to create the experiment. We think we were able to show a new method for studying human cognition. As for the HRTF using steam audio, there were still many unfinished parts, and it was difficult to say that it was an accurate 3D sound. However, since it seemed to be localized outside the head, we believe that the accuracy can be improved using HRTFs customized for each individual in the future.

Second, we were able to study human attention from the point of view of multimodal. Experiment 1 revealed that the presentation of multimodal stimuli leads reaction time to be faster than uni-modal. It was also found that the reaction time was shorter when the auditory and visual stimuli' spatial location coincided. These results can be applied to the real world. For example, when creating a UI, if we want the user to respond more quickly, it is better to stimulate multiple modalities. For example, by presenting visual and auditory stimuli near the emergency buttons, we can respond more quickly than uni-modal to avoid danger. Let us consider sirens as another example. The experiment results suggest that the warning light's spatial location and the alarm sounding's spatial location can attract people's attention more quickly if they are matched rather than separated. Ambulance sirens may disperse other driver's attention because the sound source localization is poor, and it is not clear from where it is approaching. The development of a siren with a well spatial localized sound source will lead Ambulance to run more safely. Experiment2, 3 revealed that Auditorily presented arithmetic problems can affect visual function and performance of a visual search. These results suggest the possibility that people's spatial attention can be influenced by solving math problems. For example, giving math problems to drivers could prevent accidents and falling asleep at the wheel.

Some auditory stimuli may reduce visual attention, given the increase in gaze entropy while solving auditorily presented arithmetic problems in experiment2. In particular, the fact that auditory perception and information processing without spatial source localization affects visual function suggests that even listening to music while walking may have some effect on visual attention. Experiments 2 and 3 suggest that visual attention and auditory attention influence each other in ways other than spatial attention.

Finally, we were able to analyze biometric data such as eye-gaze, head-movement, and reaction time using statistical methods. We believe that the GLMM model, which allows direct analysis of raw data rather than individual representative values, and fitting methods that do not rely solely on averages, have strengthened reaction time analysis reliability.

Acknowledgments

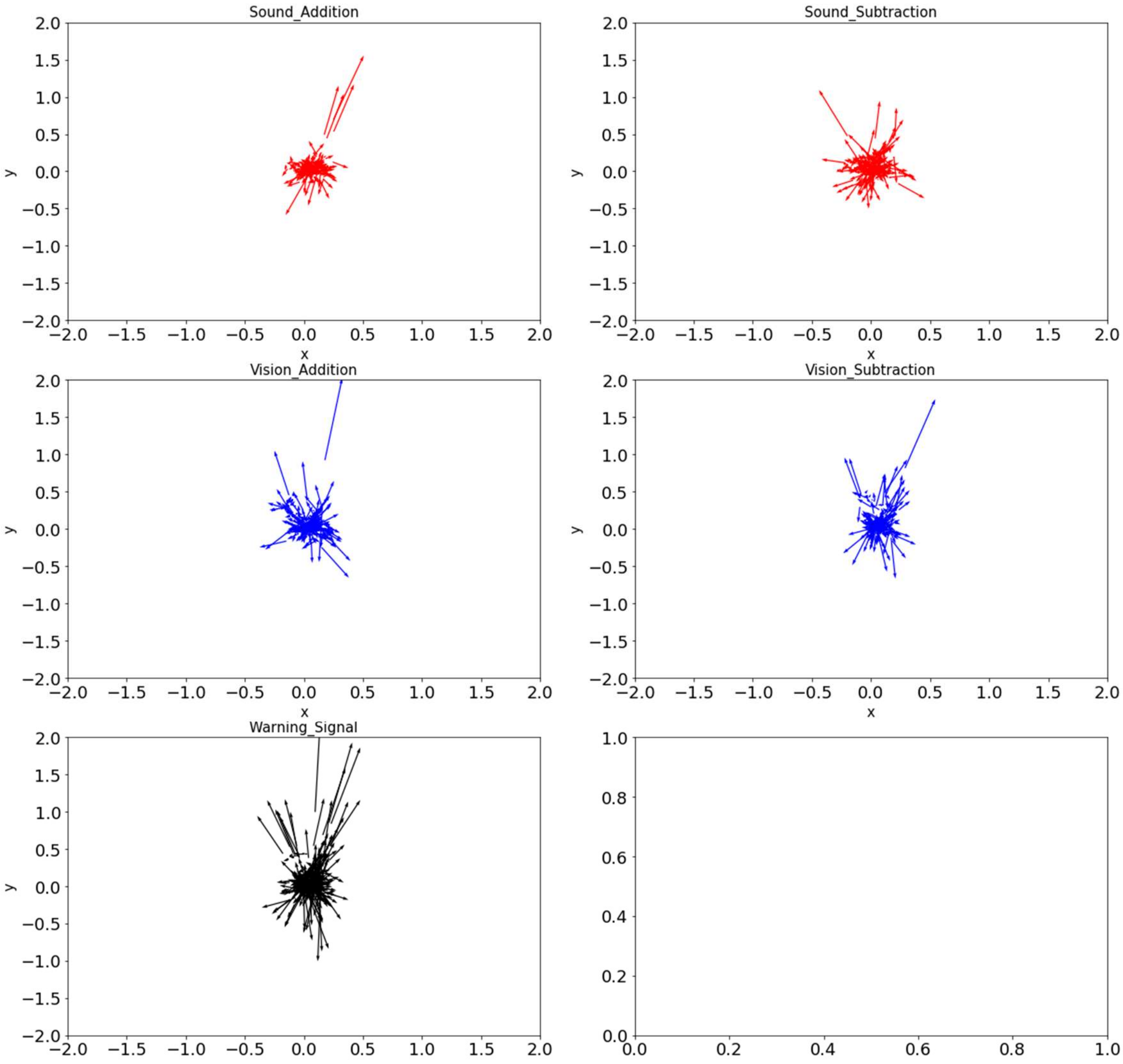
I would like to thank Professor Micheletto Ruggero for teaching me the basics of research. Without his advice, my research would not have been possible.

I would like to thank my seniors in Micheletto Laboratory (Cognitive Informatics Laboratory) for their frequent advice. In particular, Mr. Fujii, a master's student, often gave me his opinions on my research. His opinions were very pertinent, and I learned a lot from him.

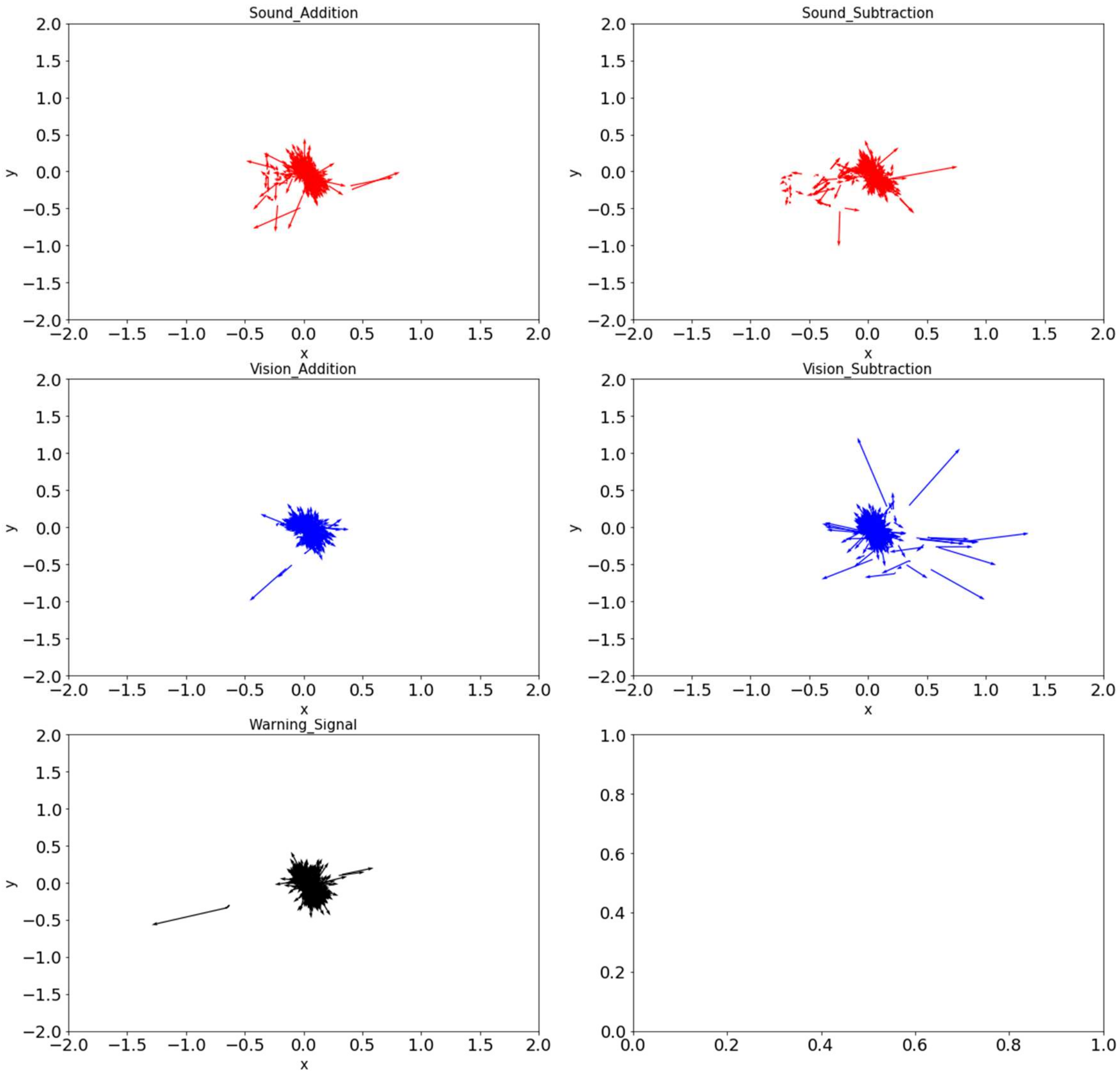
I would also like to express my deepest gratitude to the participants of the experiment. Perceptual and psychological experiments cannot be conducted without participants. Although our research activities were conducted in the midst of a new coronavirus (COVID-19) pandemic, many students from the Faculty of Sciences and the Faculty of International Liberal Arts participated in the experiment. In particular, I would like to thank Professor Mika Hirai and her seminar students for their participation in the experiments.

Appendix

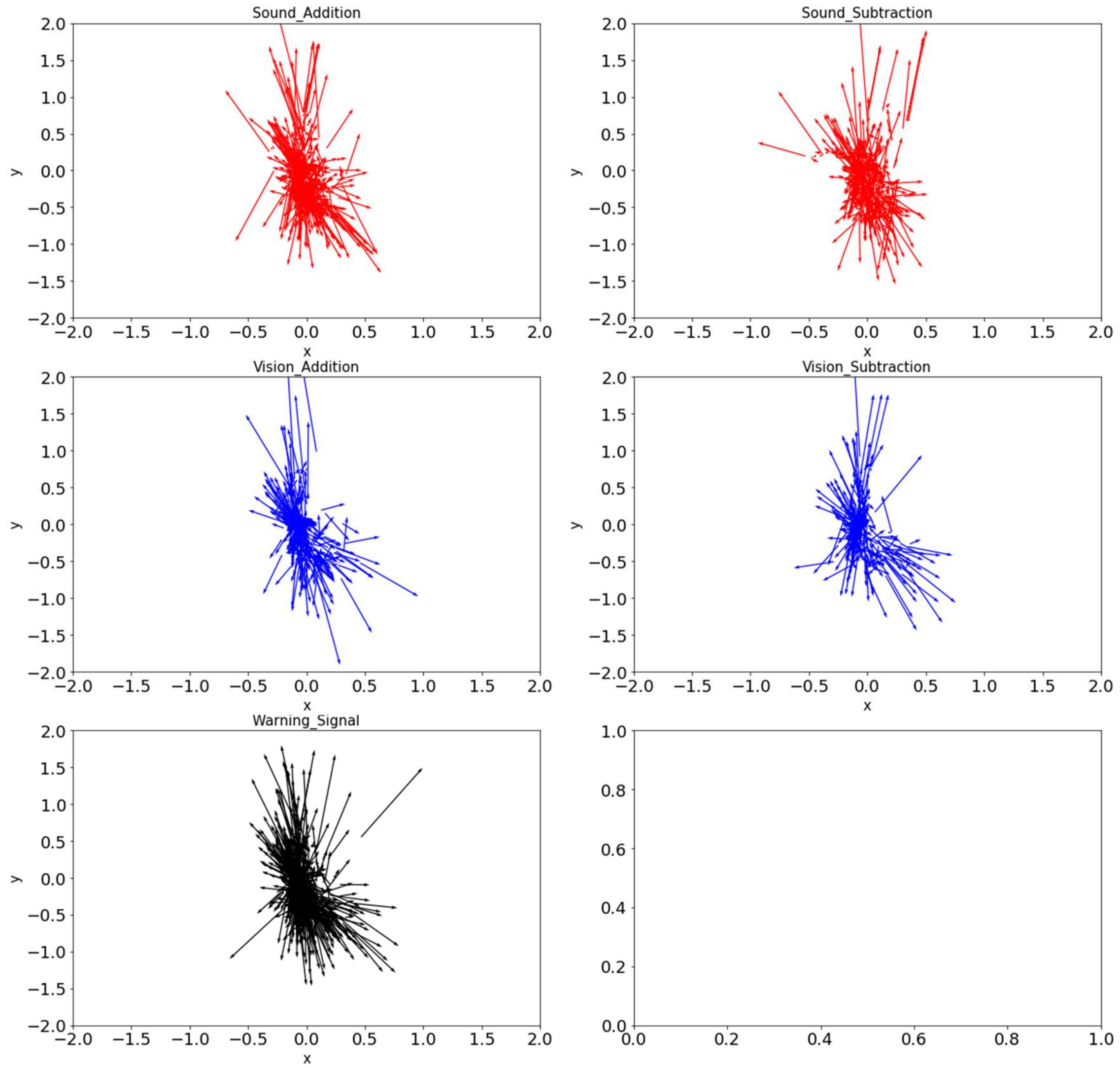
Other eight people's Vector plot in Experiment 2



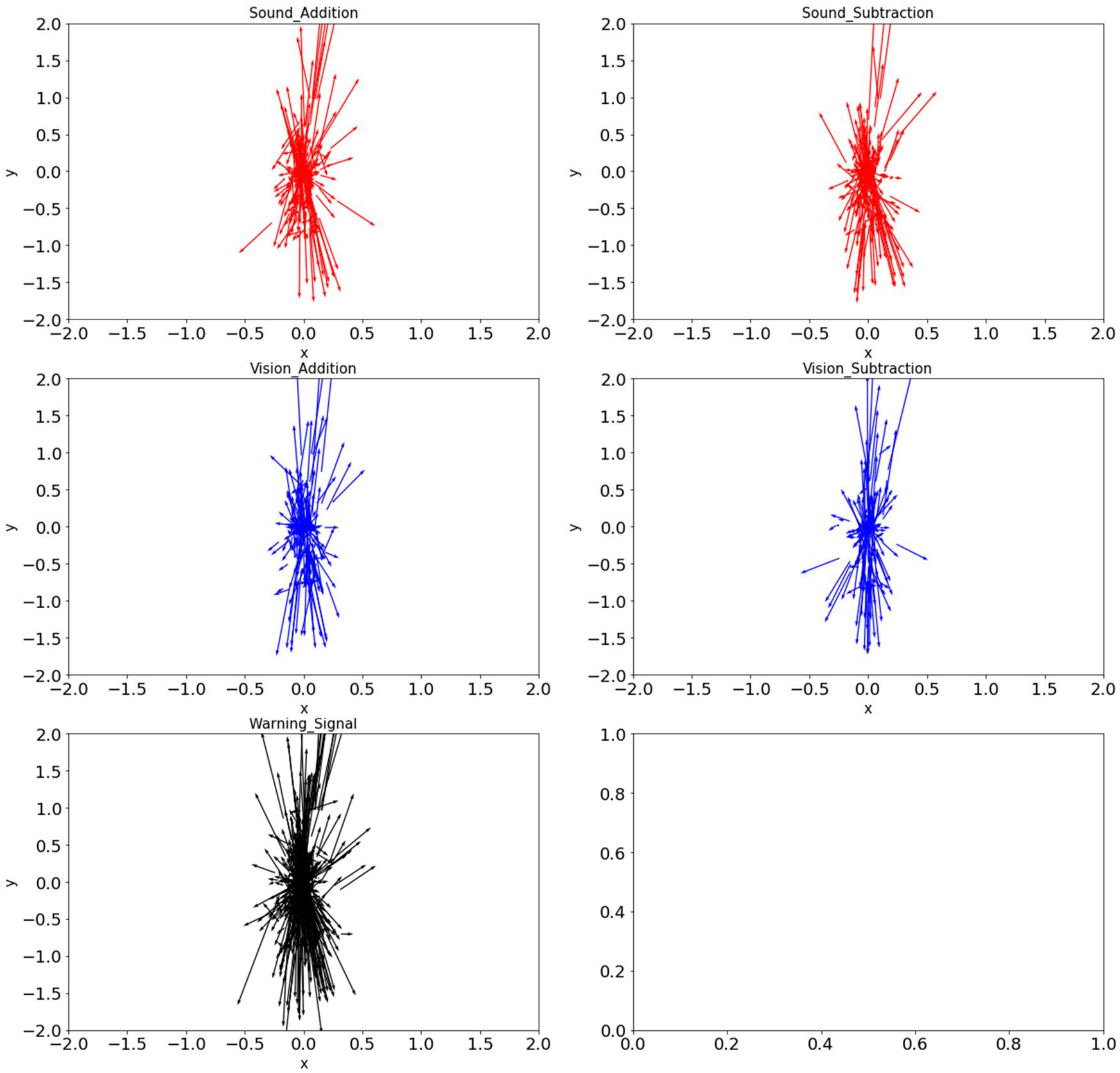
ID 1


Sound_Addition
Sound_Subtraction
Vision_Addition
Vision_Subtraction
Warning_Signal
x
y

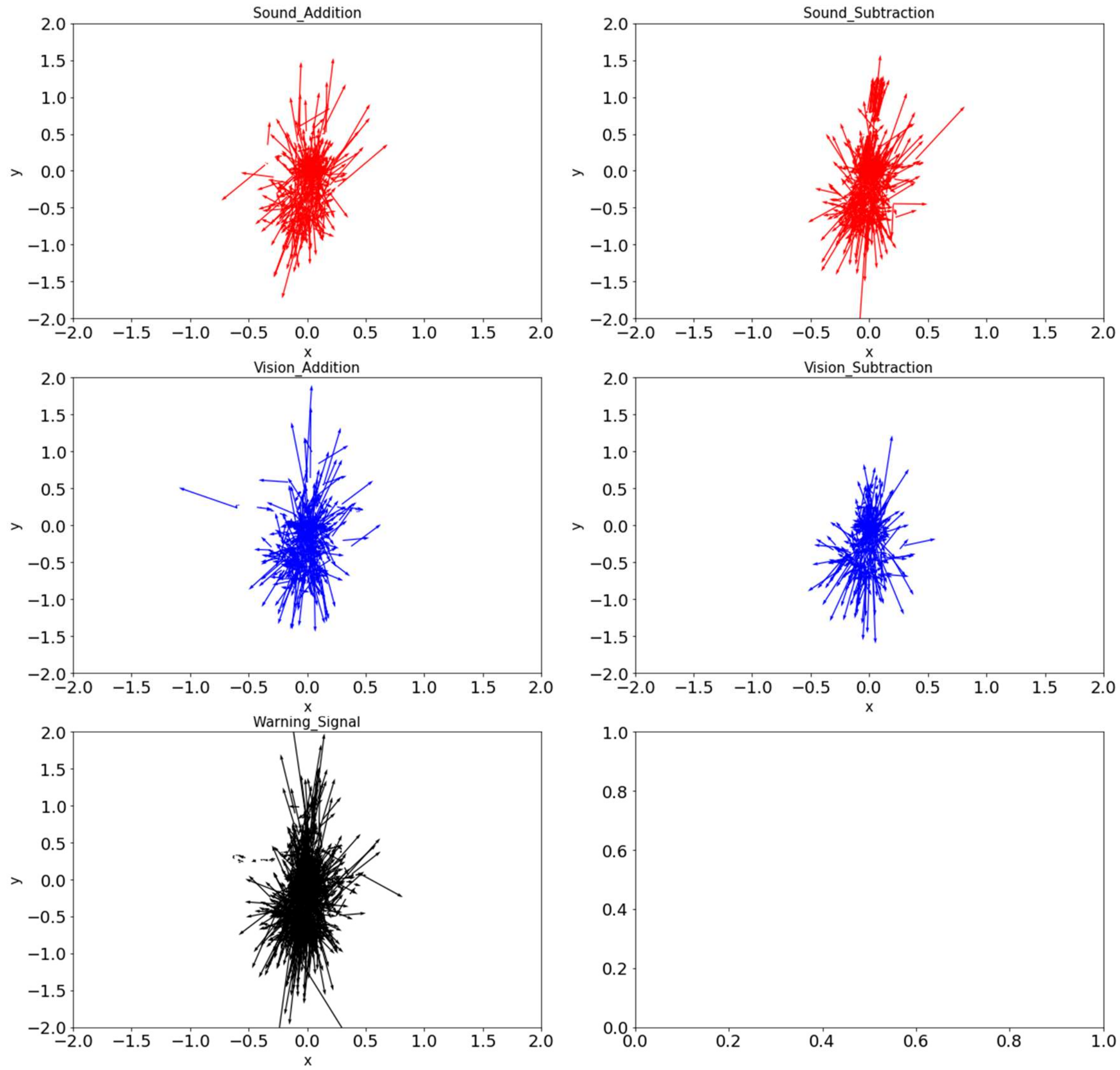
ID 4


Sound_Addition
Sound_Subtraction
Vision_Addition
Vision_Subtraction
Warning_Signal
x
y

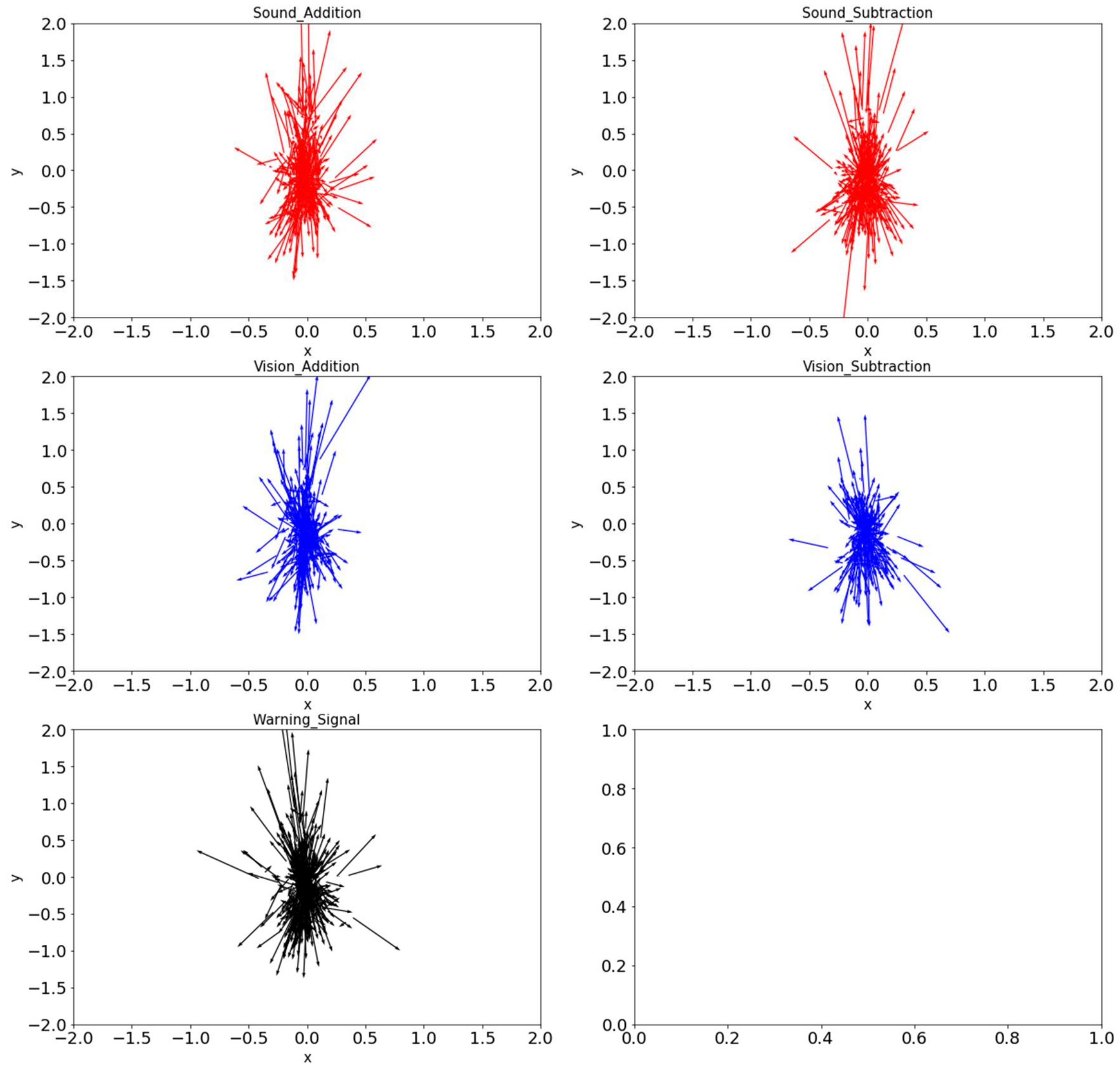
ID 6


Sound_Addition
Sound_Subtraction
Vision_Addition
Vision_Subtraction
Warning_Signal
x
y
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
0.2
0.4
0.6
0.8

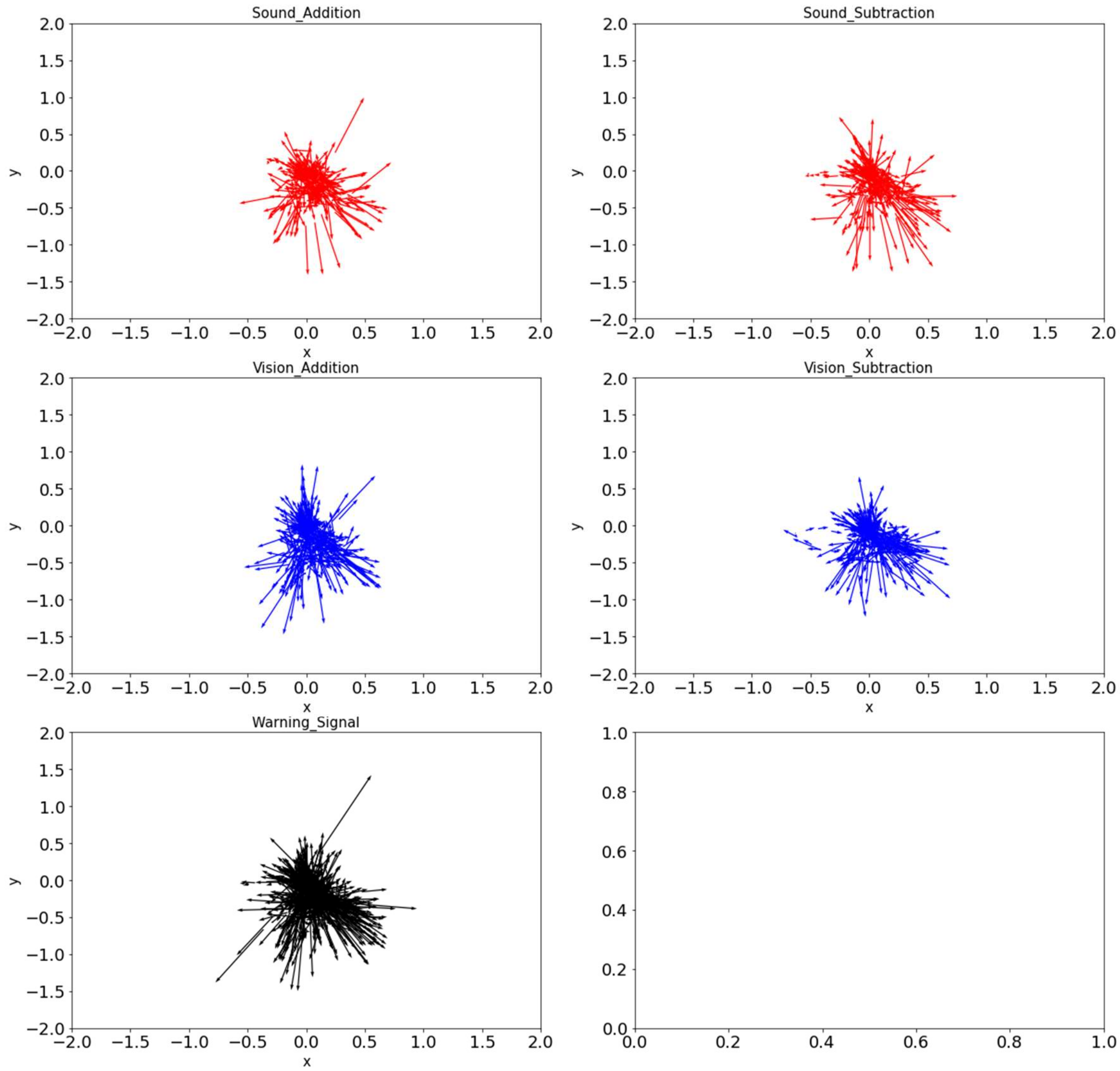
ID 7


Sound_Addition
Sound_Subtraction
Vision_Addition
Vision_Subtraction
Warning_Signal
x
y

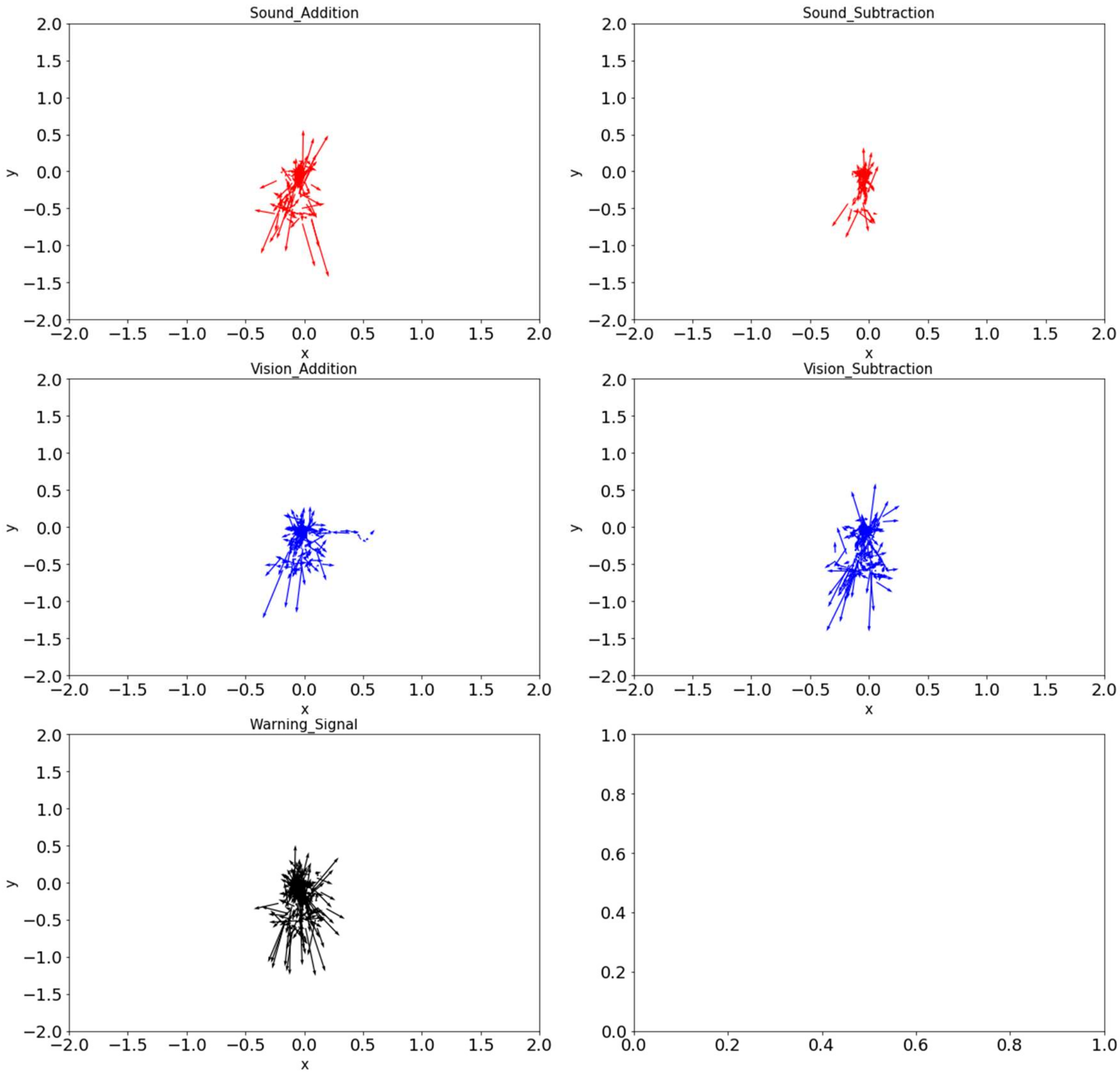
ID 8


Sound_Addition
Sound_Subtraction
Vision_Addition
Vision_Subtraction
Warning_Signal
x
y

ID 9


Sound_Addition
Sound_Subtraction
Vision_Addition
Vision_Subtraction
Warning_Signal
x
y

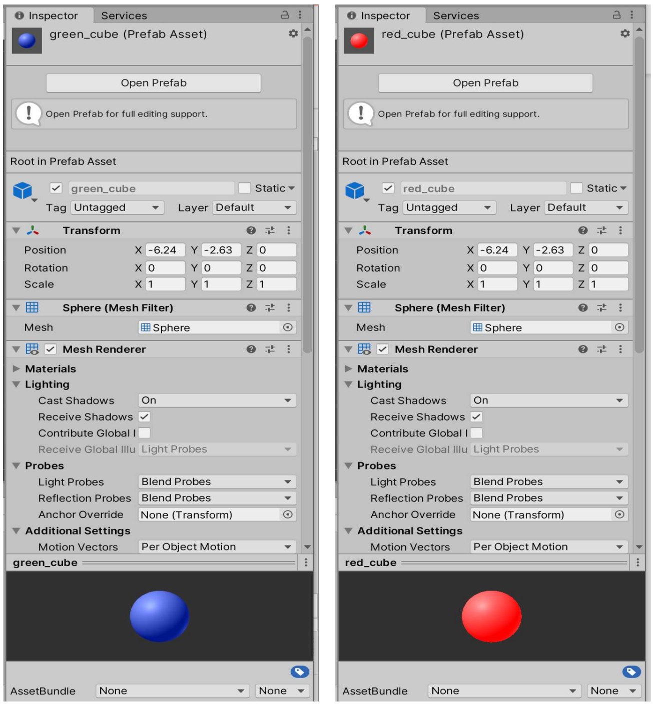
ID 10



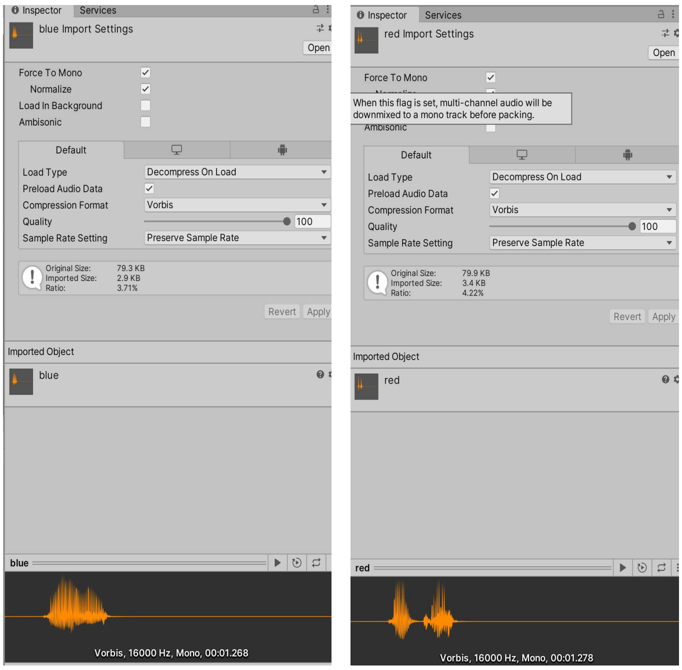
ID 12(eye camera position is variable)

Script and project scene On Unity

<u>Experiment 1</u>



Visual stimuli(Scale 1 [m]×1 [m] ×1 [m])



Auditory stimuli

Main script

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using System;

public class mainscript : MonoBehaviour
{

    //data table 用
    public static float positionX;
    public static float positionY;
    public static float positionZ;
    public int sound;
    public int i;
    //視覚刺激のオンオフを管理
    public int visual_onoff;
    public int    accord_color;
    //soundscript から変数を取得用
    public soundscript SC;
    GameObject gameobject;

    string color;


    // Start is called before the first frame update
    void Start()
    {
        //コルーチンの取得
        StartCoroutine("Coroutine");

    }

    public IEnumerator Coroutine()
    {

        //スタート合図
        Debug.Log("Start");
        //出現回数の設定
         for ( i = 0; i <100 ; i++)
        {

              //前回のオブジェクトを削除
            foreach (Transform transform in gameObject.transform)
          {
            var go = transform.gameObject;
            Destroy(go);
          }
            //プレハブ取得
              gameobject =GameObject.Find("OVRCameraRig");
            SC = gameobject.GetComponent<soundscript>();
            accord_color =SC.Getsign_accord();
            if (accord_color==0)
            {
                color = "red_cube";
            }else
            {
                color =    "green_cube";
            }

            GameObject parentObject = (GameObject)Resources.Load (color);

            //座標を soundscript から取得 or 視覚刺激なし
            visual_onoff = UnityEngine.Random.Range(0,3);
```

```
            //出現スパンの設定
            yield return new WaitForSeconds(3.0f);
            if (visual_onoff==0)//視覚刺激をパス
            {
                yield return new WaitForSeconds(4.0f);
                continue;
            }else if (visual_onoff==1)//視覚刺激を表示するために仮想音源の変数を取得
            {

              positionX =SC.GetX();
              positionY =SC.GetY();
              positionZ= SC.GetZ();
            }else
            {
              positionX =-SC.GetX();
              positionY =SC.GetY();
              positionZ= SC.GetZ();
            }

            // 生成
              GameObject childObject = Instantiate(parentObject, new Vector3(positionX, positionY,positionZ),
Quaternion.identity ) as
              GameObject;
              childObject.transform.parent = transform;
              Debug.Log(Time.time);
             yield return new WaitForSeconds(4.0f);



        }
      yield return new WaitForSeconds(3.0f);
      Debug.Log("Finish");
      //コルーチンを終了
      yield break;
    }


    // Update is called once per frame
    public void Update()
    {

    }
    //変数受け渡し
    public int Getsound_type()
    {
        return sound;
    }
    public int Getnumber()
    {
        return i;
    }
    public int Getvisual_onoff()
    {
        return visual_onoff;
    }

}
```

Sound script

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using System;
using System.Data;
using System.IO;
using System.Text;
using System.Threading.Tasks;
using System.Linq;

public class soundscript : MonoBehaviour
{
    GameObject childObject;
    GameObject gameobject;
    public mainscript MS;
    //時間計測用
    public static float elapsedTime;
    public static bool counter_flag = false;
    //data table
    public DataTable table = new DataTable("Table");
    public static bool one;
    public static float positionX;
    public static float positionY;
    public static float positionZ;
    public int which;
    public int visual_accord;
    public int left_or_right;
    public bool sign_flag;
    public string path;
    public string path2;
    public int i;
    public int correct;
    public int accord;

    public int[] V;

    public int[] V2;

    public int[] V3;


    List<List<float>> head = new List<List<float>>();



    // Start is called before the first frame update
    void Start()
    {       V = new int[100];
            for (int i_d = 0; i_d < 4; i_d++)
            {
                for (int i_m = 0; i_m < 25; i_m++)
                {
                    V[i_m+i_d*25] = i_d+1;
                }
            }
            List<int> list =new List<int>(V);
            list.RemoveAll(item=>item==0);
            V2=list.ToArray();
            V3 = V2.OrderBy(i=> Guid.NewGuid()).ToArray();
        //コルーチンの取得
         StartCoroutine("Coroutine");
         GameObject parentObject = (GameObject)Resources.Load ("Scube");
         gameobject =GameObject.Find("Steam Audio Manager Settings");
         MS = gameobject.GetComponent<mainscript>();
```

```
}

public IEnumerator Coroutine()
{   //カラムの設定
    table.Columns.Add("time", Type.GetType("System.Single"));
    table.Columns.Add("sound_type", Type.GetType("System.Int32"));
    table.Columns.Add("sound_accord", Type.GetType("System.Int32"));
    table.Columns.Add("visual_onoff", Type.GetType("System.Int32"));
    table.Columns.Add("visual_accord", Type.GetType("System.Int32"));
    table.Columns.Add("correct", Type.GetType("System.Int32"));
    //ファイルパスの設定
    path = Application.persistentDataPath + "/"    +" VR_hearTest.csv";
    path2 = Application.persistentDataPath + "/"    +" VR_hearTest_head.csv";
    using( FileStream fs = File.Create(path));
    using( FileStream fs = File.Create(path2));

    //スタート合図
    Debug.Log("Start");
    //出現回数の設定
     for (int i = 0; i <100 ; i++)
    {

        //回数の記録
         sign_flag = false;
         head.Add(new List<float> {i});
        // ゲームオブジェクトの子の Transform を列挙
         foreach (Transform transform in gameObject.transform)
        {
          // Transform からゲームオブジェクト取得・削除
             var go = transform.gameObject;
             Destroy(go);
        }
        //プレハブ取得
        GameObject parentObject = (GameObject)Resources.Load ("Scube");

         //どの種類の音刺激をだすか
         which = V3[i];
          if (which == 1) //右から音声
         { positionX = 10;
            positionZ =12;
            positionY =0;
             left_or_right = 0;//音声成分の右左
         }else if(which == 2) //左から音声
         {
            positionX = -10;
            positionZ =12;
            positionY =0;
             left_or_right = 0;//音声成分の右左
         }else if (which == 3) //右から音声
         { positionX = 10;
            positionZ =12;
            positionY =0;
             left_or_right=    1;//音声成分の右左
         }else if(which == 4) //左から音声
         {
            positionX = -10;
            positionZ =12;
            positionY =0;
             left_or_right = 1;//音声成分の右左
         }
```

```
//内容と方向は一致しているか確認（一致試行、不一致試行の確認）
   switch (left_or_right)
    {//右成分
     case 0:
              if (which==1||which==3)//右方向
              {
                   accord=0;//一致
              }else
              {
                   accord=1;//不一致
              }
          break;
     default:
              if (which==2||which==4)//左方向
              {
                   accord=0;//一致
              }else
              {
                   accord=1;//不一致
              }
          break;
}

 yield return new WaitForSeconds(3.0f);
 //visual_accord の設定
switch (MS.Getvisual_onoff())
    {
     case 0://視覚なし
              visual_accord=0;
          break;
     case 1://音声方向と一緒
              if (accord==0)//音声が一致
              {
                    visual_accord=1;//視覚一致試行
              }else//音声が不一致
              {
                    visual_accord=2;//視覚不一致試行
              }
          break;
     default://音声方向と反対
              if (accord==0)//音声が一致
              {
                   visual_accord=2;//視覚不一致試行
              }else//音声が不一致
              {
                   visual_accord=1;//視覚一致試行
              }
          break;
}

// 生成
 GameObject childObject = Instantiate(parentObject, new Vector3(positionX, positionY,positionZ),
Quaternion.identity) as
 GameObject;
 childObject.transform.parent = transform;
 Debug.Log(Time.time);
//計測初期化、開始
 elapsedTime = 0 ;
 counter_flag = true;
```

```
            //開始の合図
              sign_flag = true;
            yield return new WaitForSeconds(4.0f);

        }

  // write
    using (var writer = new CsvWriter(path2))
            {
                writer.Write(head);
            }
  ConvertDataTableToCsv(table,path,true);

  //コルーチンを終了
  yield break;
}


// Update is called once per frame
public void Update()
{ //計測
  if (counter_flag == true)
        {
            elapsedTime += Time.deltaTime;
            float vx = OVRManager.display.acceleration.x;
            float vy = OVRManager.display.acceleration.y;
            float vz = OVRManager.display.acceleration.z;
            head.Add(new List<float> {vx,vy,vz});
        }

    //入力の管理
  if (OVRInput.GetDown(OVRInput.RawButton.X))
        {
         //時間、座標の記録
          counter_flag = false;
          Debug.Log(elapsedTime.ToString());
          switch (left_or_right)
          {//左
              case 1:
                      correct=0;
                      break;
              default:
                      correct=1;
                      break;
          }
          one = true;
        }

  if (OVRInput.GetDown(OVRInput.RawButton.A))
        {
            counter_flag = false;
            Debug.Log(elapsedTime.ToString());


           switch (left_or_right){
            case 0:
                    correct=0;
                    break;
            default:
                    correct=1;
                    break;
           }

          one = true;

        }
```

```
            //リセットとアプリ終了
            if (OVRInput.GetDown(OVRInput.RawButton.Y))
             {
                    Application.LoadLevel(0);
             }

            if (one)
                {
                    DataRow dr = table.NewRow();
                    dr["time"] = elapsedTime;
                    dr["sound_type"] =    which;
                    dr["sound_accord"] = accord;
                    dr["visual_accord"] =visual_accord;
                    dr["visual_onoff"]= MS.Getvisual_onoff();
                    dr["correct"]= correct;
                    table.Rows.Add(dr);
                    one = false;
                }

        }

    //変数受け渡し
    public float GetX()
        {
            return positionX;
        }

    public float GetY()
        {
            return positionY;
        }
    public float GetZ()
        {
            return positionZ;
        }
    public bool Getsign_flag()
        {
            return sign_flag;
        }
    public int Getsign_accord()
        {
            return left_or_right;
        }
```

Csv への変換部分は（https://dobon.net/vb/dotnet/file/writecsvfile.html および https://garafu.blogspot.com/2014/10/c-list-csv.html）のコードを利用したため割愛した。

Speaker script

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class speakerscript : MonoBehaviour
{
	public soundscript SC;
	GameObject gameobject;
	public AudioSource left;
	public AudioSource right;


	// Start is called before the first frame update
    void Start()
    {
	gameobject =GameObject.Find("OVRCameraRig");
	SC = gameobject.GetComponent<soundscript>();
	AudioSource[] audioSources = this.GetComponents<AudioSource> ();


	left = audioSources[1];
	right = audioSources[0];
	int left_or_right = SC.Getsign_accord();
	switch (left_or_right)

    {
	case 0:
			audioSources[0].PlayOneShot(audioSources[0].clip);
			break;
	default:
			audioSources[1].PlayOneShot(audioSources[1].clip);
			break;
    }
    }

	// Update is called once per frame
	void Update()
	{

	}
}
```

Text Change

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using TMPro;

public class Textchange : MonoBehaviour
{

	private GameObject Text;

	public soundscript SC;
	GameObject gameobject;
	public bool S_flag;



	// Start is called before the first frame update
	void Start()
	{
	    Text = GameObject.Find("New Text");
	      gameobject =GameObject.Find("OVRCameraRig");
	      SC = gameobject.GetComponent<soundscript>();
	    Text.GetComponent<TextMesh>().text = " ";
	}

	// Update is called once per frame
```

```
    void Update()
    {
            S_flag = SC.Getsign_flag();
        if (S_flag == true)
        {
                Text.GetComponent<TextMesh>().text ="start";
        }
        else if(S_flag == false)
        {
                 Text.GetComponent<TextMesh>().text ="ready";
        }
    }
}
```

Experiment 2

Main script

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using System;
using System.IO;
using System.Text;
using System.Linq;

public class mainscript : MonoBehaviour
{
        public String name="";
        GameObject childObject;
        //時間計測用
        public static float elapsedTime;
        public static bool counter_flag = false;
        //データ保存用
        static bool one;
        public int sign_flag;
        Fove.Unity.FoveInterface fove;
        Ray right;
        Ray left;
        Ray eyes;
        //実験制御用
        public int block;
        public int i;
        public int i_0 ;
        public int i_1 ;
        public int i_2 ;
        public int i_3 ;
        public int[] blocklist;
        public int[] blocklist2;
        public int[] blocklist_to_use;
        //保存用リスト
        List<List<object>> head = new List<List<object>>();
        List<List<object>> header = new List<List<object>>();
        //関数取得用
        public speakerscript SC;
        public Fove.Unity.FoveManager manager;
        GameObject gameobject;
        GameObject gameobject2;
        // スタート時に読み込み
        public void Start()
        {
                        //game object の取得
                        gameobject =GameObject.Find("Scube");
                        //script の取得
                        fove = this.GetComponent<Fove.Unity.FoveInterface> ();
                        manager = this.GetComponent<Fove.Unity.FoveManager> ();
                        SC = gameobject.GetComponent<speakerscript>();
                        //出題順の設定
                        blocklist = new int[80];
                        for (int i_d = 0; i_d < 20; i_d++)
                {
                         for (int i_m = 0; i_m < 4; i_m++)
                         {
                               blocklist[i_m+i_d*4] = i_m;
                         }
                }
                        //array をリストに変更し順序をシャッフル
            List<int> list =new List<int>(blocklist);
```

```
        blocklist2=list.ToArray();
        blocklist_to_use = blocklist2.OrderBy(i=> Guid.NewGuid()).ToArray();
                StartCoroutine("JOB");
    }
    public IEnumerator   JOB()
    {     sign_flag = 0;
                // 変数初期化、開始
                elapsedTime = 0 ;
                i_0 = 0;
        i_1 = 0;
                i_2 = 0;
                i_3 = 0;
                //スタート合図
                Debug.Log("Start");
                //保存先の設定
                string path2 = "C:/Users/students/Downloads/VR_addsub_fixed2_"+name+".csv";
                using( FileStream fs = File.Create(path2));
                //カラムの設定
                header.Add(new                                                                    List<object>
{"right.origin.x","right.origin.y","right.origin.z","left.origin.x","left.origin.y","left.origin.z","right.direction.x","right.direction.y","ri
ght.direction.z",
                        "left.direction.x","left.direction.y","left.direction.z",

    "eyes.origin.x","eyes.origin.y","eyes.origin.z","eyes.direction.x","eyes.direction.y","eyes.direction.z","elapsedTime","i
","i_1","i_2","i_3","i_4","block","saccade","posx","posy","posz","eularAng.x","eularAng.y","eularAng.z"});



                //練習
                for ( int b = 0; b <4; b++)
                {
                        //前回の値を初期化;
                        block =0;
                        counter_flag = true;
                        sign_flag = 50;
                        //span
                        yield return new WaitForSeconds(3.0f);
                        //開始の合図
                        sign_flag = 2;
                        yield return new WaitForSeconds(0.1f);
                        //開始の合図
                        //sound の種類
                        if (b==0|b==4)
                        {
                                SC.Startspeak(40);
                                sign_flag=1;
                        }
                        else if (b==1|b==5)
                        {
                                SC.Startspeak(41);
                                sign_flag=1;
                        }
                        //sound の種類
                        if (b==2|b==6)
                        {
                                sign_flag=44;
                        }
                        else if (b==3|b==7)
                        {
                                sign_flag=45;
                        }
                        yield return new WaitForSeconds(5.0f);
                }
                sign_flag = 0;
```

```
		//五秒 ready
		yield return new WaitForSeconds(5.0f);

			for ( i = 0; i <80; i++)
			{
				//前回の値を初期化;;
				counter_flag = true;
				sign_flag = 50;
				block = 10;

				  int i_block= blocklist_to_use[i];
				//span
				yield return new WaitForSeconds(3.0f);
				//開始の合図
				sign_flag = 2;
				yield return new WaitForSeconds(0.1f);
				//開始の合図
				//sound の種類
				if (i_block==0)
				{
					block=1;
					SC.Startspeak(i_0);
					sign_flag = 1;
				    i_0 = i_0+1;
				}
				else if (i_block==1)
				{
					block=2;
					SC.Startspeak(i_1+20);
					sign_flag=1;
					i_1 = i_1+1;
				}
				//sound の種類
				if (i_block==2)
				{
					block=3;
					sign_flag=4+i_2;
					i_2 = i_2+1;
				}
				else if (i_block==3)
				{
					block=4;
					sign_flag=24+i_3;
					i_3 = i_3+1;
				}
				yield return new WaitForSeconds(5.0f);
			    sign_flag = 2;
			    yield return new WaitForSeconds(0.1f);
			}

		counter_flag = false;
		header.AddRange(head);
		//datatable を csv で記録
		using (var writer = new CsvWriter(path2))
		{
			writer.Write(header);
		}
		//終了
		Debug.Log("Finish");
		sign_flag = 3;
		yield break;

	}

	public int Getsign_flag()
```

```
{
        return sign_flag;
}

//入力待ち
IEnumerator WaitInput (){
        yield return new WaitUntil(() => Input.GetKeyDown(KeyCode.Space));
}

public　void FixedUpdate ()
{
        if (counter_flag == true) {
                //タイマー用
                elapsedTime += Time.deltaTime;
                //ヘッドトラッキング（加速度)

                //位置
                float px = this.transform.position.x;
                float py = this.transform.position.y;
                float pz = this.transform.position.z;
                //回転
                float rx = this.transform.eulerAngles.x;
                float ry = this.transform.eulerAngles.y;
                float rz = this.transform.eulerAngles.z;
                var rays = fove.GetGazeVector(Fove.Eye.Left);
                left = rays.value;
                var rays2 = fove.GetGazeVector(Fove.Eye.Right);
                right = rays2.value;
                var rays3 = fove.GetCombinedGazeRay ();
                eyes = rays3.value;

                //リスト形式で記録
                head.Add (new List<object> {
                        right.origin.x,
                        right.origin.y,
                        right.origin.z,
                        left.origin.x,
                        left.origin.y,
                        left.origin.z,
                        right.direction.x,
                        right.direction.y,
                        right.direction.z,
                        left.direction.x,
                        left.direction.y,
                        left.direction.z,
                        eyes.origin.x,
                        eyes.origin.y,
                        eyes.origin.z,
                        eyes.direction.x,
                        eyes.direction.y,
                        eyes.direction.z,
                        elapsedTime,
                        i,
                        i_0,
                        i_1,
                        i_2,
                        i_3,
                        block,
                        Fove.Unity.FoveManager.IsUserShiftingAttention().value,
                        px,
                        py,
                        pz,
                        rx,
                        ry,
                        rz
                });
        }
```

```
        }
}
```
}CSV への変換部分は割愛(実験 1 と同じサイトのコードを利用)

Speaker script

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class speakerscript : MonoBehaviour
{
    public mainscript MS;
    GameObject gameobject;
    public List<AudioClip> slist = new List<AudioClip>();
     AudioSource audioSource;
    // Start is called before the first frame update
    public void Startspeak(int sn)
    {
        //オーディオソースのセット

            audioSource =GetComponent<AudioSource> ();
             audioSource.PlayOneShot(slist[sn]);
    }
    public void Stopspeak()
    {
        //オーディオソースのセット
            audioSource =GetComponent<AudioSource> ();
             audioSource.Stop();
    }
};
```

Text change

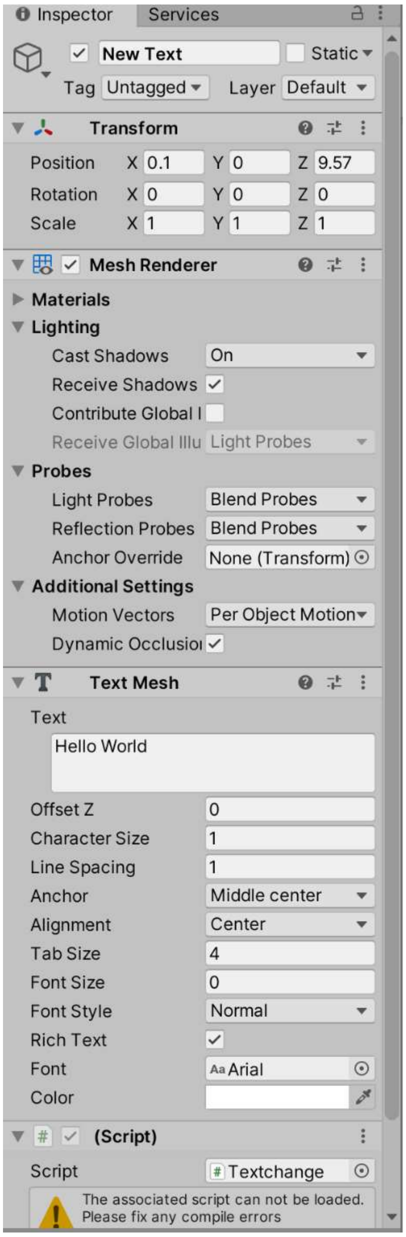
```
using UnityEngine;

public class Textchange : MonoBehaviour
{
    private GameObject Text;
    GameObject gameobject;
    public mainscript MS;
    public int S_flag;
    // Start is called before the first frame update
    void Start()
    {
        Text = GameObject.Find("New Text");
        gameobject =GameObject.Find("FoveInterface");
        MS = gameobject.GetComponent<mainscript>();
        Text.GetComponent<TextMesh>().text = " ";
    }

    // Update is called once per frame
    void Update()
    {
        //制御用変数の取得
        S_flag = MS.Getsign_flag();
        //指示の表示
        if (S_flag == 1)
        {
                        Text.GetComponent<TextMesh>().text ="000";
        }
        else if(S_flag == 0)
        {
             Text.GetComponent<TextMesh>().text ="ready";
        }
        else if(S_flag == 2)
        {
            Text.GetComponent<TextMesh>().text =" ";
        }else if(S_flag == 3)//終了合図
        {
```

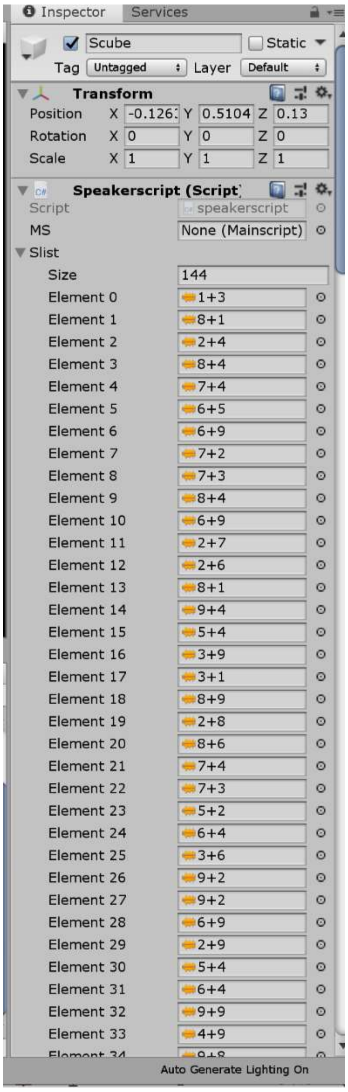
```
    Text.GetComponent<TextMesh>().text ="end";
}else if (S_flag == 4)//足し算
{
        Text.GetComponent<TextMesh>().text ="1+5";
}else if (S_flag == 5)
{
        Text.GetComponent<TextMesh>().text ="2+7";
}else if (S_flag == 6)
{
        Text.GetComponent<TextMesh>().text ="3+9";
}else if (S_flag == 7)
{
        Text.GetComponent<TextMesh>().text ="6+5";
}else if (S_flag == 8)
{
        Text.GetComponent<TextMesh>().text ="7+3";
}else if (S_flag == 9)
{
        Text.GetComponent<TextMesh>().text ="8+1";
}else if (S_flag == 10)
{
        Text.GetComponent<TextMesh>().text ="5+4";
}else if (S_flag == 11)
{
        Text.GetComponent<TextMesh>().text ="8+9";
}else if (S_flag == 12)
{
        Text.GetComponent<TextMesh>().text ="9+4";
}else if (S_flag == 13)
{
        Text.GetComponent<TextMesh>().text ="6+5";
}else if (S_flag == 14)
{
        Text.GetComponent<TextMesh>().text ="9+1";
}else if (S_flag == 15)
{
        Text.GetComponent<TextMesh>().text ="7+4";
}else if (S_flag == 16)
{
        Text.GetComponent<TextMesh>().text ="8+6";
}else if (S_flag == 17)
{
        Text.GetComponent<TextMesh>().text ="7+3";
}else if (S_flag == 18)
{
        Text.GetComponent<TextMesh>().text ="2+8";
}else if (S_flag == 19)
{
        Text.GetComponent<TextMesh>().text ="7+2";
}else if (S_flag == 20)
{
        Text.GetComponent<TextMesh>().text ="6+9";
}else if (S_flag == 21)
{
        Text.GetComponent<TextMesh>().text ="8+4";
}else if (S_flag == 22)
{
        Text.GetComponent<TextMesh>().text ="9+2";
}else if (S_flag == 23)
{
        Text.GetComponent<TextMesh>().text ="3+1";
}else if (S_flag == 24)//引き算
{
        Text.GetComponent<TextMesh>().text ="1-1";
            }else if (S_flag == 25)
            {
                        Text.GetComponent<TextMesh>().text ="4-3";
            }else if (S_flag == 26)
            {
```

```
				Text.GetComponent<TextMesh>().text ="6-2";
			}else if (S_flag == 27)
			{
				Text.GetComponent<TextMesh>().text ="3-1";
			}else if (S_flag == 28)
			{
				Text.GetComponent<TextMesh>().text ="5-2";
			}
			else if (S_flag == 29)
			{
				Text.GetComponent<TextMesh>().text ="3-2";
			}else if (S_flag == 30)
			{
				Text.GetComponent<TextMesh>().text ="7-2";
			}else if (S_flag == 31)
			{
				Text.GetComponent<TextMesh>().text ="7-3";
			}else if (S_flag == 32)
			{
				Text.GetComponent<TextMesh>().text ="6-4";
			}else if (S_flag == 33)
			{
				Text.GetComponent<TextMesh>().text ="6-3";
			}else if (S_flag == 34)
			{
				Text.GetComponent<TextMesh>().text ="7-5";
			}else if (S_flag == 35)
			{
				Text.GetComponent<TextMesh>().text ="8-4";
			}else if (S_flag == 36)
			{
				Text.GetComponent<TextMesh>().text ="8-1";
			}else if (S_flag == 37)
			{
				Text.GetComponent<TextMesh>().text ="9-4";
			}else if (S_flag == 38)
			{
				Text.GetComponent<TextMesh>().text ="8-7";
			}else if (S_flag == 39)
			{
				Text.GetComponent<TextMesh>().text ="8-6";
			}else if (S_flag == 40)
			{
				Text.GetComponent<TextMesh>().text ="9-2";
			}else if (S_flag == 41)
			{
				Text.GetComponent<TextMesh>().text ="7-6";
			}else if (S_flag == 42)
			{
				Text.GetComponent<TextMesh>().text ="9-8";
			}else if (S_flag == 43)
			{
				Text.GetComponent<TextMesh>().text ="8-3";
			}
			else if (S_flag == 50)//警告信号
			{
				Text.GetComponent<TextMesh>().text ="+";
			}else if (S_flag == 44)//練習用
			{
				Text.GetComponent<TextMesh>().text ="3+3";
			}else if (S_flag == 45)//練習用
			{
				Text.GetComponent<TextMesh>().text ="3-3";
			}
	}
}
```



Visual stimuli (Text)

Experiment 3



Auditory stimuli

Speaker script

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class speakerscript : MonoBehaviour
{
    public mainscript MS;
    GameObject gameobject;

    public List<AudioClip> slist = new List<AudioClip>();
     AudioSource    audioSource ;




    // Start is called before the first frame update
    public void Startspeak(int sn)
    {

       audioSource =GetComponent<AudioSource> ();
       audioSource.PlayOneShot(slist[sn]);
    }
    public void Stopspeak()
    {
        //オーディオソースの二重再生防止

       audioSource =GetComponent<AudioSource> ();
       audioSource.Stop();
    }
};
```

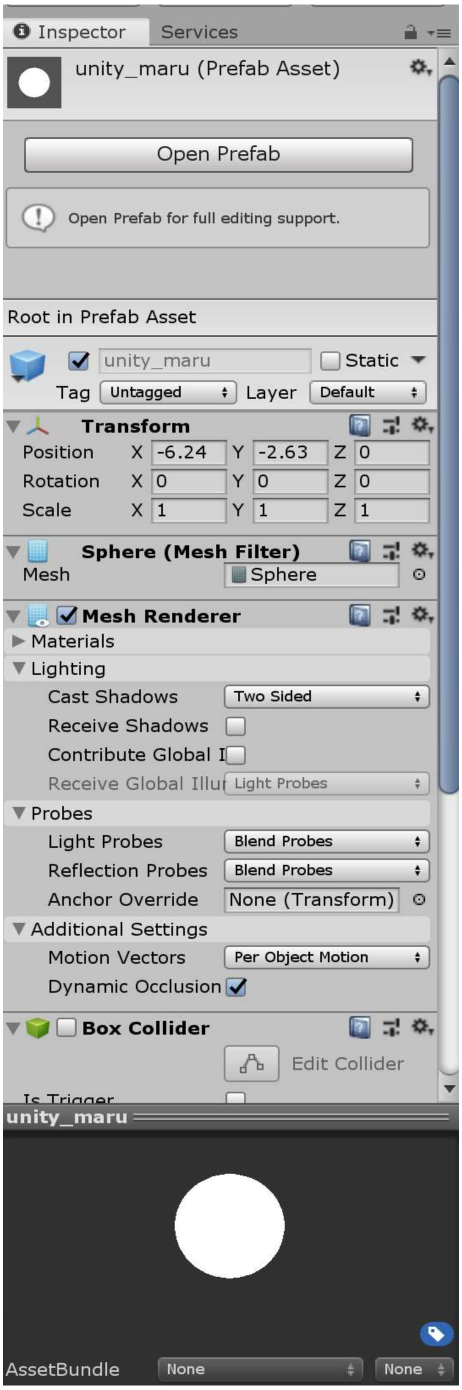
Text change

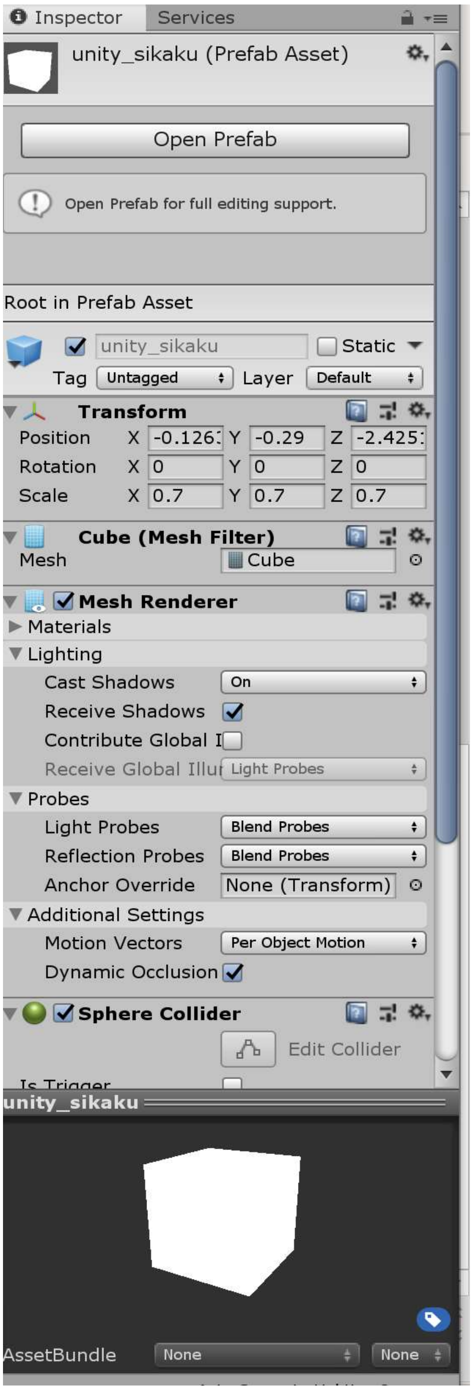
```
using System.Collections.Generic;
using UnityEngine;
using TMPro;

public class Textchange : MonoBehaviour
{
    private GameObject Text;
    GameObject gameobject;
    public mainscript MS;
    public int S_flag;
    // Start is called before the first frame update
    void Start()
    {
        Text = GameObject.Find("New Text");
        gameobject =GameObject.Find("OVRCameraRig");
        MS = gameobject.GetComponent<mainscript>();
        Text.GetComponent<TextMesh>().text = " ";
    }

    // Update is called once per frame
    void Update()
    {
        //制御用変数の取得
        S_flag = MS.Getsign_flag();
        //指示の表示
        if (S_flag == 1)
        {
            Text.GetComponent<TextMesh>().text ="start";
        }
        else if(S_flag == 0)
        {
             Text.GetComponent<TextMesh>().text ="ready";
        }
```

```
        else if(S_flag == 2)
        {
            Text.GetComponent<TextMesh>().text =" ";
        }
        else if(S_flag == 3)
        {
            Text.GetComponent<TextMesh>().text ="end";
        }
        else
        {
            Text.GetComponent<TextMesh>().text ="test is finished";
        }
    }
}
```





Cube object (scale 1 [m]× 1 [m]× 1 [m])and

Sphere object (scale 0.7 [m]× 0.7 [m]× 0.7 [m])

Main script

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using System;
using System.Data;
using UnityEngine.SceneManagement;
using System.IO;
using System.Text;
using System.Threading.Tasks;
using System.Linq;



public class mainscript : MonoBehaviour
{
    GameObject childObject;
    //時間計測用
    public static float elapsedTime;
     public static float reactionTime;
    public static bool counter_flag = false;
     public static bool reaction_flag = false;
    //データ保存用
    static bool one;
    public static float positionX;
    public static float positionY;
    public static float positionZ;
    public float angleA;
    public float angleB;
    public int sign_flag;
    public GameObject centereye;
    public Vector3 headsetPos;
    public Vector3 headsetOri;

    public DataTable table = new DataTable("Table");
    //実験制御用
    public int sound;
    public int i;
    public int i_n;
    public int nnb;
    public string prehubN;
    public int TargetN;
    //保存用リスト
    List<List<float>> head = new List<List<float>>();

    //target 指定用
    public int[] V;
    public int[] V2;
    public int[] Targetlist_a;
    //関数取得用
    public speakerscript SC;
    GameObject gameobject;

    // Start is called before the first frame update
public void Start()
{
    start2();
        gameobject =GameObject.Find("Scube");
        SC = gameobject.GetComponent<speakerscript>();
        Input.gyro.enabled = true;
        centereye =GameObject.Find("CenterEyeAnchor");
    //target リストの作成
```

```
            V = new int[72];
            for (int i_d = 0; i_d < 24; i_d++)
            {
                for (int i_m = 0; i_m < 3; i_m++)
                {
                    V[i_m+i_d*3] = i_d*10+i_m*3+2;
                }
            }
            List<int> Targetlist =new List<int>(V);
            //出現しない地点
            V2=Targetlist.ToArray();
            Targetlist_a = V2.OrderBy(i=> Guid.NewGuid()).ToArray();
}


public async Task start2()
{     sign_flag = 0;
      // 変計測初期化、開始
      elapsedTime = 0 ;
      i_n=0;
      //スタート合図
      Debug.Log("Start");
      //カラムの設定
      table.Columns.Add("time", Type.GetType("System.Single"));
      table.Columns.Add("Xaxis", Type.GetType("System.Int32"));
      table.Columns.Add("Yaxis", Type.GetType("System.Int32"));
      table.Columns.Add("Zaxis", Type.GetType("System.Int32"));
      table.Columns.Add("soundnumber", Type.GetType("System.Int32"));
      table.Columns.Add("targetnumber", Type.GetType("System.Int32"));
      //保存先の設定
      string path = Application.persistentDataPath + "/"    +" VRTest.csv";
      string path2 = Application.persistentDataPath + "/"    +" VR_hearTest_head.csv";
      using( FileStream fs = File.Create( path));
      using( FileStream fs = File.Create(path2));
      //
      int[] blockarray = new int [] {0,1,3,2};
      //五秒 ready
      await Task.Delay(5000);
      //出現回数の設定
     for (int i_block = 0; i_block < 4; i_block++)
      {
            int targetblock =blockarray[i_block];
            for ( i = 0; i <72; i++)
            {
                  //前回の値を初期化;
                  deleteObj();
                  reactionTime =0;
                  counter_flag = true;
                  sound=0;
                  //span
                  await Task.Delay(100);
                  //開始の合図
                  sign_flag = 2;
                  //sound の種類
                  if (targetblock==0)
                  {
                        sound=0;
                  }
                  else if (targetblock==1)
                  {
                        sound=1;

                  }else if (targetblock==2)
                  {
                        sound=2;
```

```
                    SC.Startspeak(i);
                }else
                {
                    sound=3;
                    SC.Startspeak(i+71);

                }

                //妨害刺激の表示
                for (int i_s =0 ; i_s < 24; i_s++)
                {
                    for (int i_b = 0; i_b < 10; i_b++)
                    {
                    //半球の設計
                     angleA=0+i_b*10;
                     angleB=0+i_s*15;
                     (float number,float number2,float number3) nc = polarconversion(10,angleA,angleB);
                     //生成
                     objectshowup("unity_maru",nc.number,nc.number2,nc.number3);
                     nnb=10*i_s+i_b;
                    }
                }

                //Target の指定
                TargetN=Targetlist_a[i];
                GameObject    TargetObj =GameObject.Find(TargetN.ToString());
                objectshowup("unity_sikaku",
TargetObj.transform.position.x,TargetObj.transform.position.y,TargetObj.transform.position.z);
                positionX=TargetObj.transform.position.x;
                positionY=TargetObj.transform.position.y;
                positionZ=TargetObj.transform.position.z;
                Destroy(TargetObj);

                //反応時間計測開始
                reaction_flag = true;

                //被験者が物体を発見しボタン入力するのを待機
                await Task.Run(() =>
                {
                    while(!OVRInput.GetDown(OVRInput.RawButton.A)){}
                });


            }

            //test 試行の終了合図
            if (i_block==0)
                {
                sign_flag=4;
                await Task.Run(() =>
                {
                    while(!OVRInput.GetDown(OVRInput.RawButton.B)){}
                });
        }
        counter_flag = false;
         //datatable を csv で記録
         using (var writer = new CsvWriter(path2))
                {
                    writer.Write(head);
                }
         ConvertDataTableToCsv(table,path,true);
         //終了
         Debug.Log("Finish");
         sign_flag = 3;
    }}
```

```
// Update is called once per frame
public void Update()
{//計測
    if (reaction_flag==true)
    {
        //反応時間取得用
        reactionTime +=   Time.deltaTime;
    }
     //入力の管理
     if (OVRInput.GetDown(OVRInput.RawButton.A))
         {
          //時間、座標の記録
           Debug.Log(elapsedTime.ToString());
           reaction_flag = false;
           deleteObj();
           SC.Stopspeak();
           one =   true;
         }
    //実験制御用（実験のリセット、アプリ終了）
     if (OVRInput.GetDown(OVRInput.RawButton.Y))
       {
               Application.LoadLevel(0);
       }
     if (OVRInput.GetDown(OVRInput.RawButton.X))
      {
              Application.Quit();
      }
    //データテーブルに計測結果を記録
    if (one){
    DataRow dr = table.NewRow();
    dr["time"] = reactionTime;
    dr["Xaxis"] =   positionX;
    dr["Yaxis"] =   positionY;
    dr["Zaxis"] =   positionZ;
    dr["soundnumber"] =   sound;
    dr["targetnumber"] =   TargetN;
    table.Rows.Add(dr);
    one = false;
    }
}
public void FixedUpdate()
{
     if (counter_flag == true)
         {
             //タイマー用
             elapsedTime += Time.deltaTime;
             //ヘッドトラッキング（加速度)
             float vx = OVRManager.display.acceleration.x;
             float vy = OVRManager.display.acceleration.y;
             float vz = OVRManager.display.acceleration.z;
             //ヘッドトラッキング（ジャイロ)
             float vrx =   OVRManager.display.angularVelocity.x;
             float vry =   OVRManager.display.angularVelocity.y;
             float vrz =     OVRManager.display.angularVelocity.z;

             centereye =GameObject.Find("CenterEyeAnchor");
             //位置
             float px =   centereye.transform.position.x;
             float py =   centereye.transform.position.y;
             float pz =   centereye.transform.position.z;
             //回転
             float rx =   centereye.transform.eulerAngles.x;
             float ry =   centereye.transform.eulerAngles.y;
             float rz =   centereye.transform.eulerAngles.z;
```

```
                //リスト形式で記録
                head.Add(new List<float> {vx,vy,vz,vrx,vry,vrz,elapsedTime,i,sound,px,py,pz,rx,ry,rz});
            }
}
     //オブジェクト生成
    public void objectshowup(string name ,float X,float Y, float Z)
    {
        //プレハブ取得
        GameObject parentObject = (GameObject)Resources.Load (name);
        // 生成
        GameObject childObject = Instantiate(parentObject, new Vector3(X,Y,Z), Quaternion.identity ) as
        GameObject;
        childObject.transform.parent = this.transform;
         prehubN=nnb.ToString();
         childObject.name =prehubN;
    }
    //極座標変換
    private (float number,float number2,float number3) polarconversion(int R ,float angA,float angB)
    {
        //極座標を座標に変換
        float number = R*Mathf.Sin(angA * Mathf.Deg2Rad) * Mathf.Cos(angB * Mathf.Deg2Rad);
        float number2 = R * Mathf.Sin(angA * Mathf.Deg2Rad) * Mathf.Sin(angB * Mathf.Deg2Rad);
        float number3 = R * Mathf.Cos(angA * Mathf.Deg2Rad);
        return (number,number2,number3);
    }
    //object 削除
    public void deleteObj()
    {
        // ゲームオブジェクトの子の Transform を列挙
        foreach (Transform transform in gameObject.transform)
        {
          // Transform からゲームオブジェクト取得・削除
          var go = transform.gameObject;
          Destroy(go);
        }

    }
    public async Task wait2seconds()
    {
        await Task.Delay(8000);
    }

   //変数受け渡し用
    public int Getsign_flag()
    {
        return sign_flag;
    }
    public float GetangleA()
    {
        return angleA;
    }
     public float GetangleB()
    {
        return angleB;
    }
    public int Getsound_type()
    {
        return sound;
    }
    public int Getnumber()
    {
        return i;
    } Csv への変換部分は割愛(実験 1 と同様)
```

References

[I] 舘暲、廣瀬通孝 (2011), バーチャルリアリティ学　日本バーチャルリアリティ学会（編），特定非営利活動法人　日本バーチャルリアリティ学会

[II] Christopher. J, Wilson and Alessandro Soranzo, The use of Virtual Reality in Psychology: A Case study in Virtual Perception, Computational and Mathematical Methods in Medicine Volume 2015, Article ID 151702, p7

[III]https://www.npa.go.jp/bureau/traffic/keitai/info.html　National Police Agency in Japan 2020/12/3 閲覧

[IV]脳科学辞典,
https://bsd.neuroinf.jp/wiki/%E7%A9%BA%E9%96%93%E7%9A%84%E6%B3%A8%E6%84%8F　2021/01/21 閲覧

[V] 熊田孝恒, 認知心理学ハンドブック　日本認知心理学会（編），有斐閣ブックス, pp.78-81

[VI] 篠原一光, 認知心理学ハンドブック　日本認知心理学会（編），有斐閣ブックス, pp.98-101

[VII] 河原純一郎, 認知心理学ハンドブック　日本認知心理学会（編），有斐閣ブックス, pp.84-87

[VIII] 藤田哲也, 認知心理学ハンドブック　日本認知心理学会（編），有斐閣ブックス, pp.138-139

[IX] 脳科学辞典
https://bsd.neuroinf.jp/wiki/%E3%83%97%E3%83%A9%E3%82%A4%E3%83%9F%E3%83%B3%E3%82%B0%E5%8A%B9%E6%9E%9C　2020/12/30 閲覧

X 藤田哲也, 認知心理学ハンドブック 日本認知心理学会（編）, 有斐閣ブックス, pp.138-139

XI Akio NISHIMURA, Kazuhiko YOKOSAWA(2012), Spatial Stimulus-Response Compatibility Effect, Japanese Psychological Review 2012 Vol55, No.4, 436-458

XII Simon, J,R.(1990). , The effects of an irrelevant directional cue on human information processing, Advances in psychology 65, 31-86, 1990

XIII Stanislas Dehaene, Serge Bossini, and Pascal Giraux (1993), The Mental Representation of Parity and Number Magnitude, Journal of Experimental Psychology:General 1993, Vol.122, No.3, 371-396

XIV Frank Restle (1970), Speed of adding and comparing numbers, Journal of Experimental Psychology 83(2p1) ,274 ,

XV Myachykov A., Cangelosi A., Fisher M.H.(2015), The oculomotor resonance effect in spatial-numerical mapping. Acta Psychologia,161,162-169

XVI 吉澤誠、北崎充晃、茅原拓郎, バーチャルリアリティ学 日本バーチャルリアリティ学会（編）, 特定非営利活動法人　日本バーチャルリアリティ学会 pp.26-39

XVII Ruggero Micheletto, 戸坂亜希, 及川虎太郎, 英語と日本語で学ぶ知覚情報科学, 共立出版 pp.38-53, pp.172-200

XVIII 日本バーチャルリアリティ学会, バーチャルリアリティ学, p33-39

XIX Ruggero Micheletto, 戸坂亜希, 及川虎太郎(2020), 英語と日本語で学ぶ知覚情報科学, 共立出版 pp38-48

XX 加藤宏, 「視覚は人間の情報入力の８０％」説の来し方と行方, 筑波技術大学テクノレポート Vol25(1) Dec, 2017

XXI 光藤宏行, 認知心理学ハンドブック 日本認知心理学会（編）, 有斐閣ブックス, p39

XXII Ruggero Micheletto, 戸坂亜希, 及川虎太郎(2020), 英語と日本語で学ぶ知覚情報科学, 共立出版, pp172-179

XXIII 和田有史, 認知心理学ハンドブック 日本認知心理学会（編）, 有斐閣ブックス p31

XXIV 和田有史, 認知心理学ハンドブック 日本認知心理学会（編）, 有斐閣ブックス p31 & (Meyer&Wuerger, 2001)

XXV 綾部早稲・井関龍太・熊田孝恒, 心理学、認知・行動科学のための反応時間ハンドブック, 126-130

XXVI櫻井美由紀, 岩崎祥一, 視覚・聴覚の二重課題における注意の偏りについての研究, 日本心理学学会第 70 回大会

XXVII Fei TIAN, Kazumitsu SHINOHARA, Yasunori KINOSADA, "Auditory accessory stimulus effect Simon effect", The 13th Annual convention of the Japanese Society for the Cognitive Psychology

XXVIII Matthias Hartmann, Fred W Mast, Martin H Fischer, Spatial biases during mental arithmetic: evidence from eye movements on a blank screen"

XXIX Yuto Yagi, Hiroshi Nittono, and Kazumitsu Shinohara "Mental addition and subtraction affect spatial attention" The 15th Conference of the Japanese Society for Cognitive Psychology.

XXX Yasuhiro Ito, Takahashi , Spatial structure of quantitative representation of numbers : Evidence from the SNARC effect", Memory&Cognition. 32(4),662-673,2004

XXXI 原田佑規・大山潤爾, バーチャルリアリティを応用した 360 度の周囲に対する基礎的注意分布の計測, Technical Report on Attention and Cognition(2019)No.2

[XXXII] Mohammad-Ali Nikouei Mahani, Karin Maria Bausenhart, Majid Nili Ahmadabadi and Rolf Ulrich, Multimodal Simon effect: A Multimodal Extension of the Diffusion model of conflict tasks”

[XXXIII] https://fove-inc.com 2020/12/20 閲覧

[XXXIV] https://fove-inc.com 2020/12/20 閲覧

[XXXV] https://support.getfove.com/hc/en-us/articles/900004261163-The-FOVE-Eye-Tracker-Every-Field-Explained 2020/12/20 閲覧

[XXXVI] http://unity.com/ja 2021/01/25 閲覧

[XXXVII] https://www1.doshisha.ac.jp/~mjin/R/Chap_16/16.html

[XXXVIII] Evan M Palmer, Todd S Horowitz, Antonio Torraiba, Jeremy M Wolfe, What are the shapes of response time distribution in Visual search,　J Exp Psychol Hum Percept Perform. 2011 Feb;(1):58-71

[XXXIX] 平井明代（2018)，教育、心理、言語系研究のためのデータ分析―研究の幅を広げる統計手法，東京図書

[XL] 植松友彦，イラストで学ぶ情報理論の考え方，講談社

[XLI] 前川黎，視覚探索での眼球運動における探索方策に関する研究，２０１９年度横浜市立大学卒業論文

[XLII] 石井俊全，大学の統計学，技術評論社

[XLIII] 東京大学教養学部統計学教室(1991)，統計学入門，東京大学出版, 242-244

[XLIV] 東京大学教養学部統計学教室,(1991) 統計学入門，東京大学出版, 237-238

[XLV] 石原正規，特集③人間工学のための計測手法　第３部：心理計測と解析（５）心理実験による計測と解析（２），

[XLVI] http://www2.hak.hokkyodai.ac.jp/fukuda/lecture/SocialLinguistics/html/12nonpara.html　2020/12/25 閲覧

[XLVII] 綾部早稲・井関龍太・熊田孝恒 (2019), 心理学、認知・行動科学のための反応時間ハンドブック, 勁草書房, p89

[XLVIII] http://cogpsy.educ.kyoto-u.ac.jp/personal/Kusumi/datasem13/hiroi.pdf 2020/12/12 閲覧 and [XLVIII]

[XLIX] RYUTA ISEKI, How have psychologists treated response time., The Japanese Journal of Psychonomic Science 2020, Vol.2, 243-249

[L] RYUTA ISEKI, How have psychologists treated response time, The Japanese Journal of Psychonomic Science 2020, Vol.2, 243-249

[LI] 綾部早稲・井関龍太・熊田孝恒(2019), 心理学、認知・行動科学のための反応時間ハンドブック, 勁草書房

[LII] http://www2.hak.hokkyodai.ac.jp/fukuda/lecture/SocialLinguistics/html/12nonpara.html 2020/12/20

[LIII] http://aoki2.si.gunma-u.ac.jp/lecture/Average/U-test.html 2020/12/20 閲覧

[LIV] 若杉晃介, 生物・社会調査のための統計解析入門：調査、研究の現場から（その５）, 農業土木学会誌, Vol72, No12, P1051-1056,a2

[LV] 橋本修二(1997), 薬理試験における統計解析の Q&A, 日薬理誌 110, 325=332

[LVI] 池田郁 (2013), 統計検定を理解せずに使っている人のために, 化学と生物, Vol51, No7,

[LVII] Toshihiko Morikawa(2008), Statistical Approaches to Multiplicity Issues in Clinical Trials, Jpn J Biomet Vol.29, Special Issue 1,

[LVIII] 荒瀬康司 (2018), 論文投稿に際しての統計的記述の留意点, 人間ドック, 33:550-570

[LIX] 池田郁 (2013), 統計検定を理解せずに使っている人のために, 化学と生物, Vol51, No7

[LX] http://www.med.osaka-u.ac.jp/pub/kid/clinicaljournalclub1.html
2020/12/16 閲覧

[LXI] ttps://support.getfove.com/hc/en-us/articles/900004261163-The-FOVE-Eye-Tracker-Every-Field-Explained

[LXII] 陰山聡, 剛体のオイラー方程式の愚直な導出, http://www.research.kobe-u.ac.jp/csi-viz/members/kageyama/docs/170110_rigid_body_euler_eq.pdf
2021/01/15 閲覧

[LXIII] 東京大学教養学部統計学教室(1991), 統計学入門, 東京大学出版会, p122

https://dobon.net/vb/dotnet/file/writecsvfile.html

https://garafu.blogspot.com/2014/10/c-list-csv.html

Using these codes to output csv file in Unity.