BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF COMPUTER SYSTEMS
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

GAME DEVELOPMENT FOR ASSESSMENT OF A PERSON'S REASONING, AUDITORY & VISUAL SKILLS

GAME DEVELOPMENT FOR ASSESSMENT OF A PERSON'S REASONING, AUDITORY & VISUAL SKILLS

BACHELOR'S THESIS
BAKALÁŘSKÁ PRÁCE

AUTHOR **ŠTĚPÁN PEJCHAR**
AUTOR PRÁCE

SUPERVISOR **doc. AAMIR SAEED MALIK, Ph.D.**
VEDOUCÍ PRÁCE

BRNO 2024

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Bachelor's Thesis Assignment

153401

Institut:	Department of Computer Systems (DCSY)
Student:	**Pejchar Štěpán**
Programme:	Information Technology
Title:	**Game Development for Assessment of a Person's Reasoning, Auditory & Visual skills**
Category:	Biocomputing
Academic year:	2023/24

Assignment:

1. Study and learn about human's reasoning, auditory & visual skills and various tests and games for their assessment.
2. Get acquainted with data management methods, machine learning techniques, and various resources related to apps and web development.
3. Find out challenges in assessment of a person's reasoning, auditory & visual skills as well as limitations of the existing methods for such assessment.
4. Select and suggest changes in design of game (and/or digital brain test) for assessment of a person's reasoning, auditory & visual skills.
5. Implement the designed game (and/or digital brain test) for assessment of a person's reasoning, auditory & visual skills.
6. Create a set of benchmark tasks to evaluate quality of the assessment of a person's reasoning, auditory & visual skills.
7. Conduct critical analysis and discuss achieved results and their contribution.

Literature:

- According to supervisor's advice.

Requirements for the semestral defence:

- Fulfillment of items 1 to 4 of the assignment.

Detailed formal requirements can be found at https://www.fit.vut.cz/study/theses/

Supervisor:	**Malik Aamir Saeed, doc., Ph.D.**
Head of Department:	Sekanina Lukáš, prof. Ing., Ph.D.
Beginning of work:	1.11.2023
Submission deadline:	9.5.2024
Approval date:	30.10.2023

Abstract

The goal of the thesis is to implement games that assess cognitive functions of the users playing them, specifically audio-visual cognitive functions and reasoning cognitive functions. The nature of the thesis is a pilot study of a bigger project. The first part of the thesis is the theoretical research. It explains what these cognitive functions are, how they work, and how we can assess them. The second part of the thesis talks about the actual game. It explains the game design and the implementation. The game was implemented in Unity using C# and a Firebase database. This part also talks about how the game assesses and presents the assessment of the cognitive functions to the users. The last part of the thesis deals with the testing of the game. The game was tested on fifteen users. Their answers are presented and evaluated in the final part.

Abstrakt

Cílem této bakalářské práce je implementovat hry, které zhodnotí kognitivní schopnosti uživatele, který je hraje. Konkrétně se jedná o schopnosti rozhodovací, audio a vizuální. Tato bakalářská práce je pilotní studie rozshálejšího projektu. První část práce obsahuje teoretický výzkum. Popisuje zmíněné kognitivní schopnosti, jak fungují a jak je můžeme hodnotit. Druhá část popisuje samotnou aplikaci. Popisuje návrh her, a jejich implementaci. Aplikace obsahující hry byla implementována v Unity v jazyce C#. Pro databázi byl použit Firebase. Tato část také vysvětluje jak hry provádí zhodnocení kognitivních schopností, a jak toto zhodnocení prezentují uživateli. Poslední část bakalářské práce popisuje testování aplikace. Aplikace byla testována na patnácti uživatelích. Jejich odpovědi jsou zdokumentovány a zhodnoceny v poslední části bakalářské práce.

Keywords

game, audio, visual, audio-visual, reasoning, test, cognitive functions, assesment, user

Klíčová slova

hra, audio, vizuální, audiovizuální, rozhodování, test, kognitivní funkce, evaluace, uživatel

Reference

PEJCHAR, Štěpán. *Game Development for Assessment of a Person's Reasoning, Auditory & Visual skills.* Brno, 2024. Bachelor's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor doc. Aamir Saeed Malik, Ph.D.

Rozšířený abstrakt

Cílem této bakalářské práce je implementace her, které hodnotí kognitivní schopnosti uživatele, jenž je hraje, konkrétně pak audiovizuální a rozhodovací schopnosti. Bakalářská práce je pilotní studie rozsáhlejšího projektu. Kognitivní rozhodovací schopnost je schopnost člověka učinit nejlepší rozhodnutí v dané situaci a na základě předem daných informací. Nejrozšířenější definice rozhodování je deduktivní (logické) rozhodování. Podle této definice se člověk snaží v každé situaci dospět k co nejlogičtějšímu výsledku. Toto se často znázorňuje pomocí sylogismu. Sylogismus je dvojice výroků, která implikuje výrok třetí. Příkladem může být tento známý sylogismus: "Všichni muži jsou smrtelní, Sokrates je muž, a proto je Sokrates smrtelný". Vidíme, že na základě dvou informací obsažených v prvních dvou výrocích můžeme dedukovat výrok třetí. V reálném životě nejsou výroky na základě kterých se rozhodujeme tak jednoznačné a dokonce může existovat více správných logických řešení.

Jelikož je rozhodování psychologický koncept, je poměrně složité ho testovat. Tento koncept je také velmi rozsáhlý, a dal by se rozdělit na podčásti. Hra implementovaná v této práci se soustředí na toleranci risku.

Audiovizuální schopnosti můžeme testovat snadněji. Existuje mnoho testů, které se dají použít. Tato práce několik z nich popisuje. Reakční testy, jsou testy, které měří reakční čas na určitý podmět. Tyto testy mohou měřit jak audio, tak vizuální schopnosti. Dále máme testy, které měří kolik objektů je člověk schopen sledovat zároveň.

Z audio testů máme kromě reakčních i testy, které hodnotí jak dobře je člověk schopen analyzovat audio stopu. Toto se dá měřit tím způsobem, že testovanému člověku přehráváme více audio stop zároveň. Testovaný člověk musí přehranou zprávu identifikovat přes ostatní audio stopy. Příkladem může být rozpoznat větu, která je spuštěna zároveň se šumem, nebo hudbou. U audiovizuální hry bylo cílem tyto testy přeformovat do zábavné hry.

Druhá část práce se zabývá návrhem a implementací. Hlavním cílem návrhu je, aby hra byla zábavná a chytlavá. Člověk, kterého hra baví a soustředí se na ni, bude mít z testování přesnější výsledky, než člověk, kterého hra nudí a hraje ji jen kvůli svému vyhodnocení. Chytlavá hra pomůže k získání více dat od uživatele a tím i k zlepšení přesnosti výsledků. Celá aplikace má minimalistický vzhled. Důvodem je to, že by aplikace měla být intuitivní, aby ji mohlo využívat co nejvíce lidí.

Audiovizuální hra je pojatá zejména jako reakční. Uživateli jsou zobrazovány různé tvary, na které musí uživatel zmáčknout dokud nezmizí. Dále jsou uživateli přehrávány audio stopy, které říkají: "left", nebo "right". Audio stopy mohou být přehrány i zároveň. V tu chvíli co je audio stopa přehrána musí uživatel zareagovat zamáčknutím správného tlačítka v jednom ze spodních rohů obrazovky.

V rozhodovací hře je cílem pomocí tří tvarů, které mají různé hodnoty, odhadnout výslednou sumu takovým způsobem, aby nebyla přestřelena definovaná suma. Hra má určitou podobnost s populární karetní hrou blackjack. Herní okno obsahuje tři tvary, každý má nad sebou interval celých čísel. Uživatel má určitý čas na zvolení počtu tvarů, které se do výsledné sumy budou počítat. Při počítání výsledné sumy je vybrána náhodná hodnota z intervalu nad tvarem. Uživatel tedy až do ukončení kola neví, jaká jeho výsledná suma ve skutečnosti bude. K rozhodování mu pomáhá text, který zobrazuje jeho maximální minimální a střední možnou sumu.

Uživatel má možnost se zaregistrovat, nebo hrát hru jako host. Při hraní jako host však nemá přístup k úplnému zhodnocení. Při prihlášení se uživatel přesune do hlavního menu. V něm má na výběr buď spustit audiovizuální hru, rozhodovací hru, otevřít kartu s instrukcemi, otevřít statistiky nebo se odhlásit či hru vypnout.

Pokud uživatel otevře okno statistik, zobrazí se mu několik grafů, které obsahují jeho výsledky. Na vrchu okna se statistikami si také může přečíst jeho percentil a jeho pořadí mezi ostatními hráči v dané hře.

Implementace hry byla provedena v Unity v jazyce C#. Pro databázi byl použit Firebase.

Hra byla testována na patnácti uživatelích. Výsledky testů odhalily, že pro většinu uživatelů byla aplikace považována spíše za hru, než test. Většina uživatelů by také doporučila hru někomu jinému. Všichni uživatelé, kteří hru hráli se založeným účtem odpověděli, že se cítili velmi dobře zhodnoceni. Uživatelé, kteří hru hráli jako hosté odpovídali na otázku střední hodnotou na škále 1-10. Tento výsledek byl očekáváný, jelikož hosté nemají přístup do statistik a jediné zhodnocení, které dostávají je obecné zhodnocení hned po ukončení hry.

Bakalářská práce dokázala, že měřit kognitivní schopnosti lidí pomocí her je možné. Testovaní useři vnímali aplikaci spíše jako hru než test, a většina z nich odpověděla, že by si hru znovu zahrála. Zároveň pro ně byla dotačující i míra, s jakou bylo provedeno jejich zhodnocení a ani jeden user neodpověděl, že by mu nějaká měřená statistika chyběla.

Game Development for Assessment of a Person's Reasoning, Auditory & Visual skills

Declaration

I hereby declare that this Bachelor's thesis was prepared as an original work by the author under the supervision of Mr. doc. Aamir Saeed Malik Ph.D.

.....
Štěpán Pejchar
May 7, 2024

Acknowledgements

I would like to thank my thesis supervisor doc. Aamir Saeed Malik Ph.D. for guiding me throughout the whole process of creating the thesis from the beginning of the school year until the finish.

Contents

List of Figures

Chapter 1

Introduction

Game development for the assessing of the cognitive skills of a person is an interesting topic. Using games for teaching a person is a concept also discovered by John Amos Comenius, a Czech bishop who devoted his life to teaching children and philosophy. His goal was to reform the pedantic educational ways into a new system tailored for the children. It is a concept proven to work. With time moving forward, people are now realizing, that games can have a different form than Comenius had in mind, computer/mobile games. It has been proven that games can be used to train cognitive functions, which the thesis references later. The question is, can games be used to assess the cognitive functions of a person? The facts about Comenius come from this source [21].

The thesis deals with reasoning and auditory and visual functions. First, we have to dive into the theory of these functions and find out what they actually are.

Carl Jung was the first to examine this topic, and the first one to define cognitive functions in his book „Psychological Types“. The following paragraphs are based on this book [14].

He said that „they are particular mental processes within a person's psyche that are present regardless of common circumstances“. He also defined the first 4 cognitive functions: thinking, feeling, sensation, and intuition. In a sense, all of these four are connected to reasoning. Reasoning is a cognitive function that helps a person resolve a situation in the best way possible. It is a skill that people have since they were born, or at least at some level.

Auditory and visual skills are a fundamental set of cognitive functions. Auditory skills describe one's ability to perceive and process information through sound. On the other hand, visual cognitive functions allow a person to perceive and process what they see.

The main idea is that assessing one's cognitive skills as a person based just on how well they see or hear is not a very good assessment for everyday life. Getting the information is a different thing than the ability to use the information. Because of this, a lot of new projects on assessing or training the cognitive skills of a person are emerging. Most of these assessment tools are in the form of tests or quizzes. The main objective of this thesis is to discover if it is possible to assess these skills in a more engaging way such as a game, and also if this type of assessment is accurate.

This thesis first addresses the theory of cognitive skills. It shows how cognitive skills work, and how people have been training and assessing them so far. It compares the methods of the assessment and examines how accurate they are, and also how interesting they are for a person trying to assess their own cognitive functions.

The main goal of this thesis is to design and program a game that is based on the outcomes of the theoretical parts. The thesis also tries to find out what can be done differently for a better assessment of these skills and to use that in the design. Also, a big part of the thesis is testing the game. The outcome should show, that games can be used in a reasonable way like assessing the cognitive functions of a person. And that the results may be more accurate because of the interest of the tested candidate.

Chapter 2

Cognitive Skill: Reasoning

The first pillar of this thesis is reasoning. It is very important to understand how reasoning works in order to assess it correctly. This chapter explains how reasoning works from a psychological standpoint, it looks into research that has already been done on this topic and analyzes it. It explains what could be done better for the assessment and designs a way to improve what has already been done. The outcome should be to design a way to assess reasoning as accurately as possible.

2.1 What is reasoning

Reasoning is one of the fundamental cognitive skills of a person. Overall, it can described as a process that uses critical thinking and logic to solve problems, make decisions, and draw conclusions in the best way possible. Reasoning is a process that happens in the frontal lobe of the brain. The frontal lobe is the largest section of the human brain. It is situated right in the front behind the forehead. Its main functions are:

- decision making
- abstract thinking
- creativity
- motor tasks
- personality expression
- social appropriateness

It can be seen that the functions of this part of the brain are both logical (like decision-making, motor tasks, and social appropriateness) and instinctive (like creativity, and abstract thought). Reasoning can also be both logical and instinctive. As Charles S. Pierce said „Instinct and reasoning complement each other as cognitive skills“. Reasoning is not a thing that is based purely on logic, that is one of the reasons why it is not so easy to assess. This paragraph was inspired by [9] and quoted [18].

Reasoning can then be divided into logical and instinctive reasoning. Logical reasoning is the form that most people use in most situations. Instinctive reasoning mainly comes into play when people are under stress or try to decide on situations that do not have any logic behind them, that is why logical reasoning is the most frequent form of reasoning a person practices.

2.1.1 Logical (deductive) reasoning

Deductive reasoning is the type of logical reasoning that offers the strongest support among scientists. The main idea is, that a person going through the reasoning process knows all the premises that lead to the decision. It ensures that the decision will be correct or in other words, there will be the most logical outcome. It means that if all the premises leading to the decision are true, and we know that they are true, it is impossible for the outcome to be false. This can be well explained with a syllogism. [1].

Syllogism is a logical argument that proves a point based on true premises. It is best explained in this well-known example: „All men are mortal. Socrates is a man. Therefore, Socrates is mortal." We know that the first two facts are definitely true which concludes that the last statement has to be true as well. As Charles S. Pierce wrote: „An explanation is a syllogism of which the major premise, or rule, is a known law or rule of nature, or other general truth; the minor premise, or case, is the hypothesis or retroductive conclusion, and the conclusion, or result, is the observed (or otherwise established) fact. "

This can also be called transitive inference. We can picture it as a transitive relation where R(a, b) and R(b, c) conclude R(a, c). It can be used to assess the abilities of a person and to use previous knowledge and logic to determine a missing piece of information. This topic is better described in the section How can we assess reasoning? What is interesting about this way of reasoning, is that a person is basically creating new information based on some facts that they were given. In the syllogism „all men are mortal", no one states that Socrates is mortal. It is a new deduction, that is created by the person who analyzes the first two sentences.

Reasoning as described earlier happens in the frontal cortex, where both rational, logical, and creative, abstract functions occur. Logical reasoning is mainly a rational process, but there are things, that can affect this rational state, for example, errors in the input data, or sleep deprivation. Let's say that the syllogism previously defined does not say that „all men are mortal" but it says that „some men are mortal". The input data would then say, that „some men are mortal", and „Socrates is a man". When analyzing this set with logical reasoning we would not be able to tell if Socrates is mortal or not because we were not given enough data to analyze.

Another thing that could happen is, that the input data would state: „All men are immortal, Socrates is a man, therefore Socrates is immortal". In this case, the transitiveness works. The deductive reasoning process concludes the logical outcome. But the outcome is still wrong.

Now in this simple example, a person would be able to identify, that the first statement is incorrect, but this is not always the case. People often conclude the deductive reasoning process with statements, that can be faulty, and they do not realize it. The process itself runs correctly, but the data is incorrect. This could be interpreted as a semantic error in one's code. The code itself is running correctly, but the outcome is incorrect.

Another thing that can affect the logical reasoning process of a person is their mental state, which is often closely related to their physical state. Let's say that a person is really tired, it is a physical state of the body, the body has no energy, but this affects the mental state of a person in a way that affects reasoning. The lack of sleep affects the frontal cortex, where reasoning happens. According to the article referenced below, the lack of sleep slows down the thought process. It leads to lower alertness and concentration, which

[1] https://en.wikipedia.org/wiki/Syllogism

is fundamental when it comes to reasoning. It makes tasks that require logical reasoning more difficult. This paragraph was based on this article [16].

Let's examine once more the syllogism example for deductive transitive reasoning. In this example, the statements are correct and the relation is transitive. If the candidate who concludes the reasoning process is sleep-deprived, there is a bigger chance of him misinterpreting the correct data because he lacks concentration. For example instead of reading, „all men are mortal", he could read, „all men are immortal" and not realize his mistake. The final conclusion of the reasoning process would be that Socrates is immortal, which is an incorrect conclusion. The paragraphs in this section were based on these sources [18] [5].

2.1.2 Understanding reasoning

As mentioned earlier logical reasoning is mainly a rational process. It is a mental practice that helps a person arrive at a conclusion. It happens in a way that is purely logical.

We could visualize this as a tree graph where there is only one correct leaf node, the most logical answer. In an ideal world, the nodes in each layer would be ordered from least logical to most logical from left to right, if a person goes through the process of reasoning, they would jump through the graph through the less logical decisions and end in the most right leaf node that would be the correct most logical answer.

The way that we can order the nodes in the tree can be based on some input data. Let's say, that the decision process is binary in a way, that we always decide between two statements as visualized in the tree. We would have some input data which could be statements of a syllogism. Let's use this syllogism: „All animals need water, a dog is an animal, therefore a dog needs water".

Each layer would describe one input information. The first node in the first layer would then state: „All animals need water", and the second node would state: „All animals don't need water". The tree always expands only the decision you make, which in an ideal case is the most logical (marked with the green color in figure 2.1).

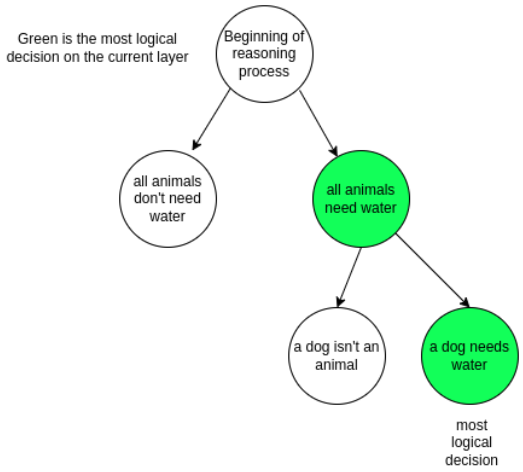

Figure 2.1: Visualization of ideal reasoning logical tree

As mentioned in the previous section, the reasoning process can be affected by errors in the input data or sleep deprivation. Let us say that we interpret incorrect data as correct,

in the tree the change would be seen as the nodes switching places. For example, if the statement was: „All animals don't need water" the graph would look like figure 2.2.

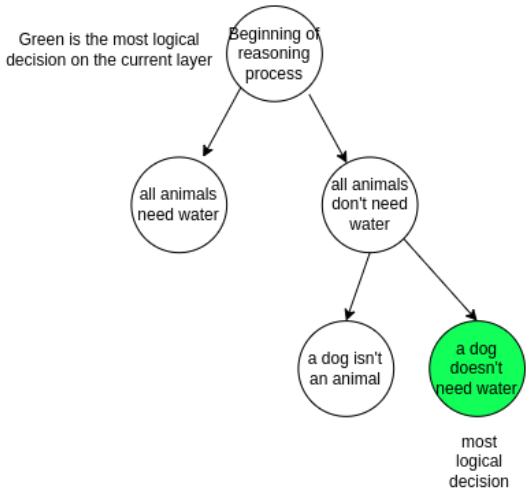

Figure 2.2: Visualization of reasoning logical tree with wrong data interpretation

Another thing that could happen with the data is that we would choose a different node, as the most logical output. Let us say that we make the most logical decision based on a probability coefficient.

In an ideal world, the chance that we pick the most logical decision would be 100%. The probability coefficient can go down depending on some external factors, like sleep deprivation. The numbers used are hypothetical and not actual. They are used just to show how the reasoning process works.

It is reasonable to assume that being tired takes down the probability coefficient from 100% to 80%. The sum of the probability coefficient in one layer of the graph has to add up to 100% so the chance of deciding the left node would be 20%. In this case, if we ran this experiment multiple times, the candidate would make the illogical decision 20% of the time.

The tree graph for a decision process adding the probability coefficient, where the candidate chose the wrong node would look like figure 2.4. In this case, the probability that the final node would be chosen is defined by the multiplication of the probability of each node in each layer. In this case, the outcome has a 0.04% chance of actually happening, the probability of the final decision would look like this.

$$0.2 \times 0.2 = 0.04\% \tag{2.1}$$

The problem with logical reasoning is, that we can only make one mistake to end up with a less logical decision. In this example, where there is just one logical answer, we can see this effect in place.

In reasoning processes that we go through every day, there can be more than 2 nodes in each layer, and there can also be more logical answers than one. Some of the decisions are much more logical than others, but that does not make the decision absolutely incorrect as in this syllogism.

As mentioned in the visualization, the probability coefficient has to sum up to 100% which means, that the more nodes in each layer there are, the less likely we are to end

up with the most logical decision. The more decisions we have to choose from, in other words, the complexity of the reasoning process affects the chance to arrive at the most logical decision. We can express this by the probability difference between N nodes, and N+1 nodes, where N is the amount of nodes. We can express this as a function with the domain of a function $D(f) =< 1; \infty >$.

$$y = \left(\frac{1}{N} - \frac{1}{N+1}\right) \tag{2.2}$$

We can see that with one node, the probability is 100% but as we add nodes, the probability gets way smaller and approaches 0 (picture 2.3).

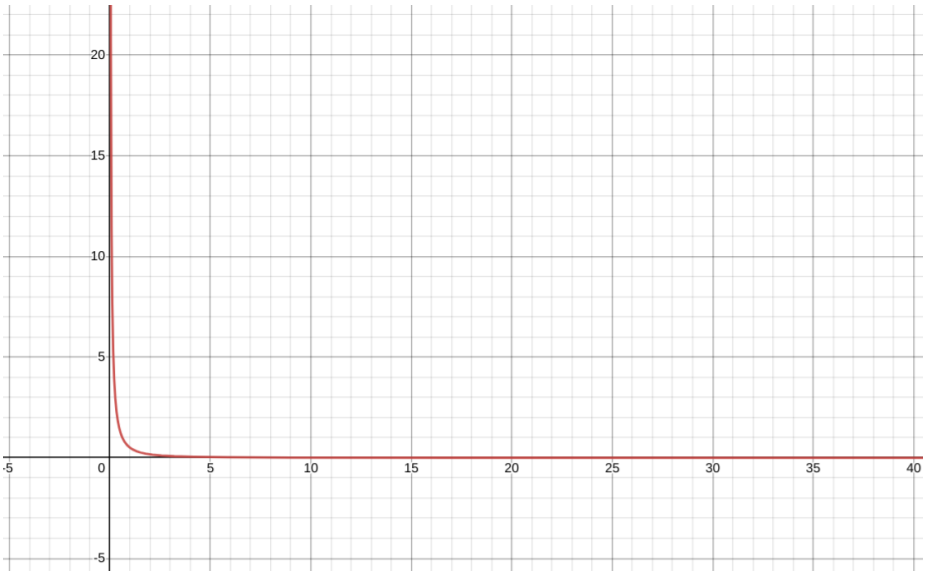

Figure 2.3: Probability decrease with the addition of nodes

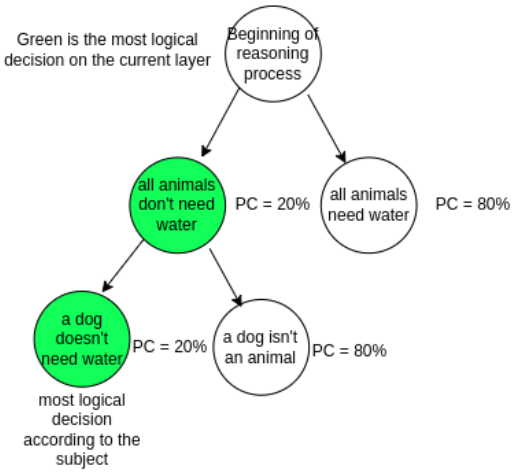

Figure 2.4: Visualization of reasoning logical tree PC

This section is based on basic probability calculations and the description of reasoning in these sources [5], [16].

2.2 How to assess reasoning

Assessing reasoning can be a tricky concept. It has already been described what reasoning is and how it can be perceived. The next part is about the concept of assessing one's reasoning ability. It also contains some taxonomies on how to define reasoning so we can better assess it.

Reasoning can be important for many situations and assessing it can be very useful. A common example is hiring an employee. Reasoning is among the set of skills that every manager should look for in an employee. It is an overall quality that helps the employee solve problems in the workspace without needing much help, understanding problems correctly and quickly, and can even give the employee an advantage in social conduct. The question is how to assess it correctly.

This is often done with some kind of test. That is because assessing someone's reasoning skills purely through observation of their actions proves challenging. By designing a test correctly we can test all the unique compartments of the reasoning ability to give an overall estimate and assessment.

In addition to it, we can camouflage the test as a game. This helps the candidate not feel so stressed about the assessment and also it creates a better environment for the candidate. That is because the person feels more natural, to handle some reasoning-related situations in a game than answering questions on how they would react in a certain situation.

All the taxonomies have some common components, which are similar or the same across different taxonomies, and some of them have distinctive components, that are specific to the taxonomy.

2.2.1 Different taxonomies for assessing reasoning

Blooms Taxonomy

The following paragraphs are based on Blooms taxonomy [2]. Reasoning has been studied since the 1950s when the first taxonomy (Bloom's taxonomy) was defined. Bloom's taxonomy is actually more of a framework to categorize educational goals. The cognitive part of Bloom's taxonomy consists of six components:

- Knowledge
- Comprehension
- Application
- Analysis
- Synthesis
- Evaluation

These components are considered common since it is the first taxonomy.

Knowledge asks, how much the candidate can remember, that is why it has been changed to remembering in the revised version referenced here [17]. It is important for the assessment

of reasoning because a person can often be in a situation, where the reasoning process is based on some former knowledge as described in the chapter above.

Comprehension is a way of actually understanding the facts. According to Bloom, it is not only important to know the right answer, but also to know why the right answer is correct. We can demonstrate this in an example.

Let us say we have a test where we ask a child in school what is $10 * 10$. Now some children memorize the whole multiplication table, so they would know, that the answer is 100. Some children on the other hand learn the process of multiplication, and even though they might remember the correct answer, they can be sure it is correct because they know how they got to the answer.

Comprehension also opens up the door to being able to answer all the questions of a similar topic without remembering the answer. The child that memorizes the multiplication table from 1 to 10 knows the answers to all the multiplications ending by 100. The child who learns how multiplication works can theoretically answer all the multiplication questions without remembering all the answers. That's why comprehension is an important part of reasoning according to Bloom.

The next part is application. A person has to be able to apply his knowledge and understanding of the situation for him to be able to decide correctly.

Analysis is very important. It shows us the ability of the candidate to deconstruct the main problem into smaller problems. We can picture this with the graphic interpretation of reasoning. Above we can see that the more nodes exist, the harder it is for the candidate to choose the correct answer. The ability to analyze lets a person transfer some nodes into another layer. They can then solve more layers (smaller problems) more easily resulting in getting to the correct answer.

Synthesis could be viewed as the opposite of analysis. While synthesising we are taking the knowledge we have gained and using it as parts of a new structure. Synthesis allows us to create new ideas and new outcomes during the reasoning process. This is very important because the outcomes to pick from are not always clear.

The last common element of Bloom's taxonomy is evaluation. Evaluation is the ability to defend opinions and ideas. It helps the person validate the outcomes they came up with and define the best ones.

Robert J. Marzano's and John S. Kendall's taxonomy

The following section is based on Robert J. Marzano's and John S. Kendall's taxonomy [20]. This is one of the newer taxonomies that came to light in 2001 and was revised in its second version in 2007. This taxonomy stands on 3 common components:

- Knowledge
- Comprehension
- Analysis

And three new distinctive components:

- The process of knowledge retrieval
- Knowledge utilization

- Self-System

Marzano and Kendall describe the process of knowledge retrieval as bringing knowledge from your long-term memory to the working memory. According to them, it is innate for every person and extremely important in the process of reasoning. That is because the knowledge we may have can mean nothing if we are not able to use it properly.

They present this idea as an example. Assume that an individual hears the following information as a part of a discussion with someone:

„The two young girls, Mary and Sally, saw the book of matches and immediately began thinking of games to play. By mid-afternoon, the house was engulfed in flames."

They say that in a logical sense, this message is incomplete. Our brain automatically adds information to make this information logical. A message that could be added by our brain might be: „They set the house on fire while playing with matches" or „They decided to play with the matches". This is what they call knowledge retrieval, and it is important for assessing the ability of reasoning.

The next distinctive component is knowledge utilization. It is a process that describes the ability of a person to finish a task that requires specific knowledge. For example, we can have an engineer using Bernoulli's principle to design a new aircraft or in programming a programmer using the knowledge of formal languages to design a new programming language. They say there are four parts to knowledge utilization:

- decision making
- problem solving
- experimenting
- investigating

The last distinctive component is the Self-System. We can describe this as some kind of motivation. It is a set of emotions, attitudes, and beliefs that determines whether or not the individual will participate in the given task and how much energy and enthusiasm he will put into it. If an individual has all the abilities to solve a problem, yet zero motivation to actually do the solving, it is as good as if he knew nothing.

The Dwyer, Hogan, and Stewart's integrated framework for Critical Thinking

The following section is based on the The Dwyer, Hogan, and Stewart's integrated framework for Critical Thinking [8]. This framework is a very important one because it is one of the newest frameworks handling the problem of critical thinking. It also consists of some common components:

- Knowledge
- Comprehension
- Analysis

And one distinctive component:

- Metacognition - self-regulatory functions

Metacognition can be described as „knowledge concerning one's own cognitive processes and products or anything related to them; and the active monitoring, consequent regulation and orchestration of these processes“ at least that is how John H. Flavell, an American developmental psychologist described it in this source[12]. This is similar to Marzano-Kendall's Self-System.

Metacognition is the ability of a person to recognize their cognitive process, the self-regulatory functions of an individual tell us how much he is invested in the action. It is the ability of the individual to recognize the importance of their actions. The more value or importance one gives to their actions or decisions, the better they should be according to this framework.

2.2.2 Existing games accessing reasoning skills

The following four paragraphs are based on the information found on the MConsulting website [28]. Some games assessing these skills already exist. A lot of these games are related to hiring employees and the hiring process. The MConsulting Prep service which designed games that assess your cognitive skills including reasoning defines two categories of these games.

Category number one is designed as a group of single games with each game assessing a specific skill. Their design called Pymetrics consists of 12 different games. Some of the skills they assess include risk tolerance, effort, decision-making, and attention. This is designed to assess separate skills more precisely. When we have a test for assessing just one cognitive function, we can evaluate the tested candidate with a different evaluation for each skill.

The second design is scenario-based games. They measure the whole set of skills by putting candidates in various contexts for example their game „survival mode“ or „business situation“. This design gives an overall evaluation, so it is not so precise. The advantage is that it is more similar to the actual world. In everyday scenarios, we often have to use more cognitive skills than just one. That is exactly what the scenario-based game does.

The games are also designed in a way so that the candidate does not know, what is tested. They just provide him with a set of rules, therefore he plays the game more naturally without thinking about the tested skill. We can see this in MConsulting's balloon game (figure 2.5). In this game, the candidate is told, that he can pump up a balloon with a click. For each pump, he receives some fictional reward. The balloon is programmed to explode at a random time. If the balloon pops, there is no reward. That is all the candidate is told. The game is designed to test risk tolerance and decision-making abilities, yet the candidate is provided only with the rules mentioned above.

Another example of scenario-based games is the McKinsey´s Solve game this paragraph is based on the information found on the McKinsey website [33]. This game is designed to test one's critical thinking, decision-making, meta-cognition, situational awareness, and systems thinking. The game is designed as several minor games that overall test these qualities. The evaluation from the game takes into account not just the results, but also the mouse clicks, and mouse movements.

More examples of a scenario-based game are games from The Talent Games company the following paragraph is written based on information found on The Talent Games company website [34]. They designed their games so that they relate to hypothetical yet real situations an employee can come across in the specific workspace. The interesting thing about this game design is, that the correct answers are decided by the employer. The evaluation then compares the answers to the ones expected by the employer. This game

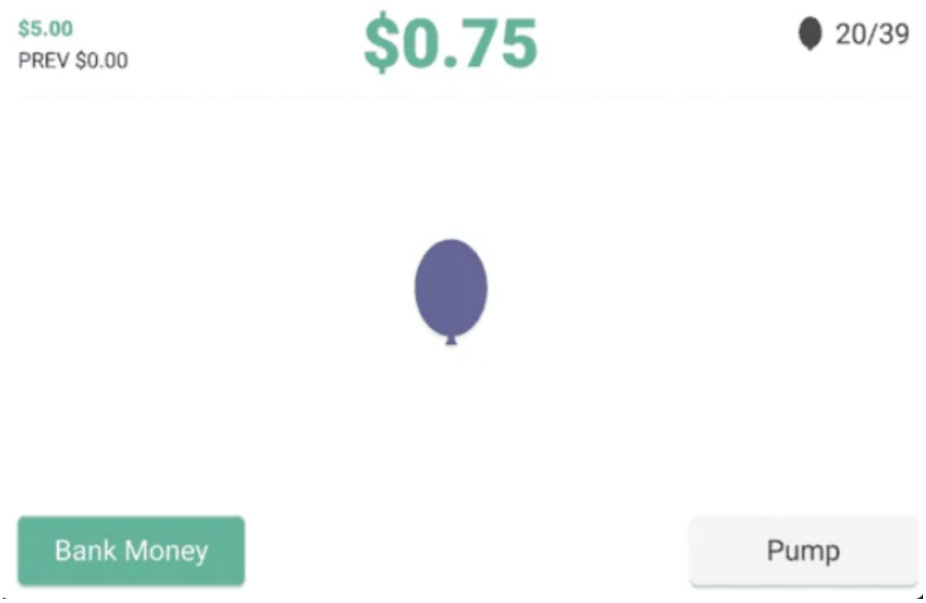

Figure 2.5: MConsulting balloon pop game

design might not be viable for the thesis game, but something could be taken from it, for example the person doing the assessment could say, that some aspects of reasoning are more important to him than others.

Some of these assessment games were played, and it has been found, that similarly to the audio-visual examples, they often resemble tests and not games. Even though these tests could be made into games very easily. Often the only difference between a test and a game is the appearance. Making assessment tests look and feel like games can have a positive impact on the tested candidate's attention. As we all know, it is harder to focus on finishing a test than playing a game, even though both of them can have the same outcome, the assessment.

Another important thing is that with the game getting more attention from the candidate, the assessment can then be more accurate. As mentioned a few of these tests/games were played, and often they were quite long, consisting of a lot of separate tests/parts. The longer the test took the worse the attention got resulting in a worse result, but that may mean a worse assessment. In table 2.6 we can see a comparison of these games, these games are private and or behind a paywall so they have not been tested. The comparison is made purely on the information available. The goal is to try and design a game, that definitely feels like a game. A game that a person would play even if it would not assess anything, basically just a catchy way to fill someone's time, but also add the feature of assessing one's cognitive abilities as precisely as possible.

2.2.3 Conclusion

Reasoning is a wanted skill. It is an innate skill that all people share. It is one of the cognitive skills of a person. Reasoning can be useful for everyday situations and should be an important skill to look for in an employee by every employer.

Games	Pymetrics - MConsulting	Scenario-based - MConsulting	Scenario-based - McKinsey solve game
assessment of one cognitive function at a time	yes	no	no
assessment of multiple cognitive functions at a time	no	yes	yes
one game	no	no	yes
minigames	yes	no	yes

Figure 2.6: Comparison of games that assess reasoning

Many services can be found online to assess an employee's reasoning skills, which can show us the real-life demand for this specific skill, one of the examples is this source [11]. It is a process that uses critical thinking, logic, and knowledge to arrive at the best conclusion.

Reasoning happens in the frontal lobe where both rational functions, like decision-making, and irrational functions like creativity are processed. One of the ways reasoning can be described is logical (deductive) reasoning. It has been defined, that reasoning is based on the amount of knowledge the person has, on the problem, and on how the person perceives the knowledge. The reasoning process also depends on how many options a person has for the outcome.

The taxonomies that describe reasoning are all connected through Bloom's taxonomy, the first taxonomy. The main pillars usually being:

- Knowledge
- Comprehension
- Analysis

The newer taxonomies also add an important point, self-regulatory functions, or self-awareness in reasoning.

For the assessment of reasoning in this thesis, some of the newer taxonomies were chosen to design a way to involve the amount of motivation the person has for the reasoning process.

The following two articles were based on these two sources [3], [4]. One thing that has not yet been described, mainly because not a lot of studies on the matter exist, is the idea of time limits in reasoning assessment. Most of the reasoning tests found had time constraints, but not a lot of studies say why. We can read, that some studies have been done on the matter, while the outcomes were often different. Some studies have found, that the difference between time-limited and time-unlimited reasoning tests is very small, and some studies find the correlation between mental speed and reasoning is quite significant. That is why time constraints will be a part of the game design.

Time limits can also help with motivation as can be read in several studies on time-motivational deadlines, an example is referenced above. Time can help in assessing the motivational aspect of reasoning. The hypothesis on the matter is, that the less time the person has for the game/test, the more motivated they will be. The longer the game takes the less motivation the person will have, and the reasoning decisions will get worse. Overall the game should test how people use received knowledge, how well they can analyze problems, and how solving a specific problem will help them solve a similar one in the future, to test comprehension.

Chapter 3

Audio-visual cognitive functions

Audio-visual cognitive skills are a skill set that is also very important and innate to almost every person. Assessing audio-visual skills is a bit more straightforward than reasoning. With reasoning being a psychological concept, it is harder to describe if a person is good at it or not. On the other hand, the audio-visual cognitive function can be assessed quite precisely. It is easier to assess if a person sees/hears something and test how well these two are connected.

These skills are often described as one set because the brain processes them together. A simple example confirming that this is true is the difference between the speed of light and the speed of sound. If our brain wouldn't account for this difference, it would be very hard for us to assess where sound is coming from at a given moment. This is described in this article [24]. The audio-visual cognitive skills are also a set that may be easier to test with a game. The game can be designed very straightforward, which can increase the ability to test these skills.

3.1 What is the audio-visual cognitive skill

The audio-visual cognitive skill is a type of multi-sensory integration. Multi-sensory integration tells us, that some perceived information, from different sensory modalities, like sight and sound are interconnected. A lot of studies have been done on this topic, for example, one of the earlier ones is a study from 1967. The following paragraphs are based on this study[23].

In this study, it is said, that the perception of sound and sight is intertwined. In the third chapter, it is presented with an example of describing an animal. They say, that it would be easier for us, to use visual information, for describing an animal, until someone asks us what the animal sounds like. Then it states, that it would in fact be possible, to describe the animal's sound purely on the visual information we get from it, for example, based on the shape of his mouth and head even though it might not be a very accurate description. This tells us how connected the audio is to the visual.

We can even imagine this in an everyday example. People often say they hear or smell pictures. This is a well-known phenomenon called synesthesia. The synesthesia description can be read in this source [1]. A specific type of synesthesia important to us is called ideaesthesia. Ideaesthesia is the specific phenomenon of hearing pictures. For example, look at these pictures (figure 3.1 and 3.2), can you hear them in your head?

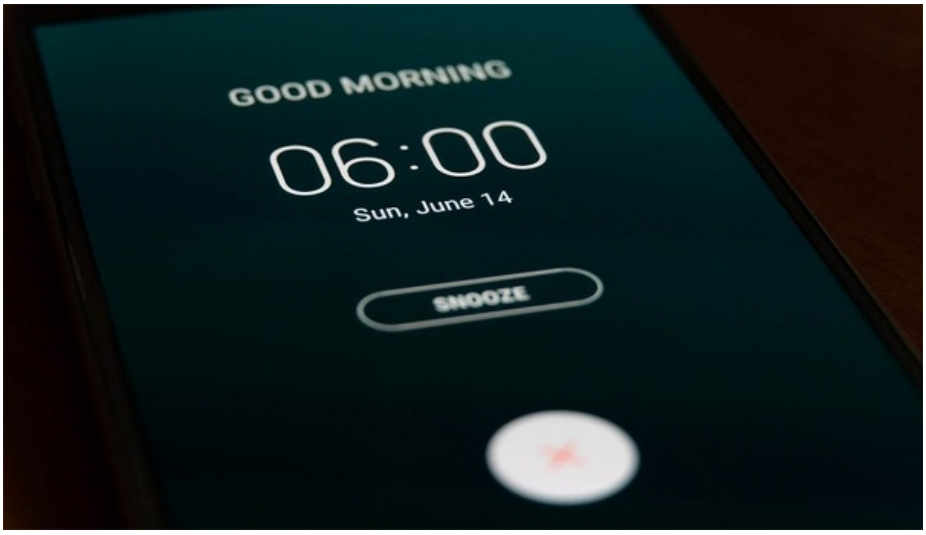

Figure 3.1: Ideaesthesia example - alarm clock

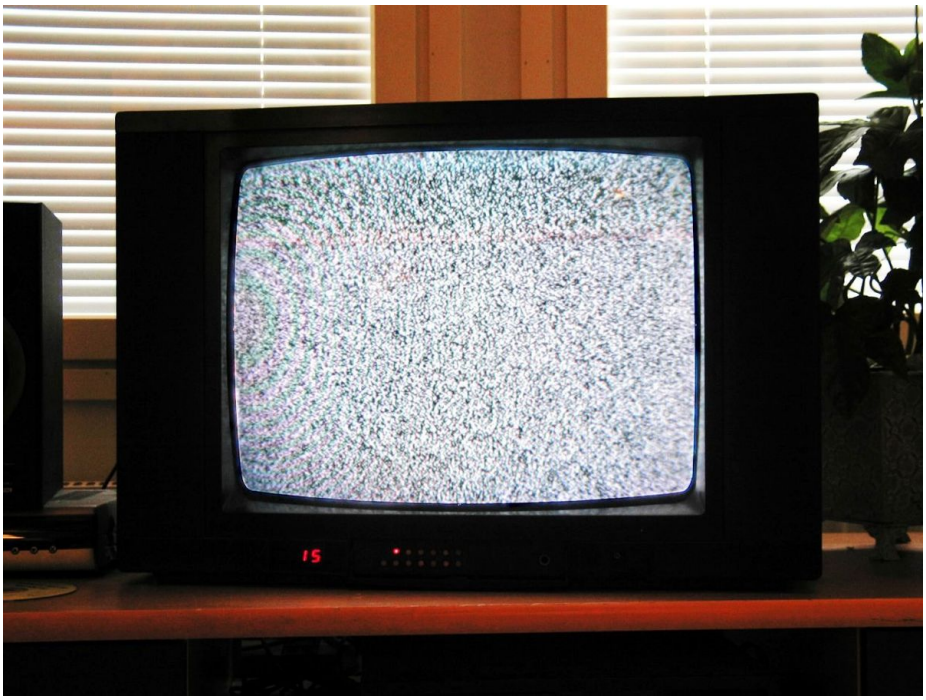

Figure 3.2: Ideaesthesia example - TV static

The following five paragraphs are based on this study [10]. Another one of these examples is the Kiki-Bouba effect. Could you try and name the two shapes in the picture (figure 3.3)? Which one is called Kiki and which one is called Bouba?

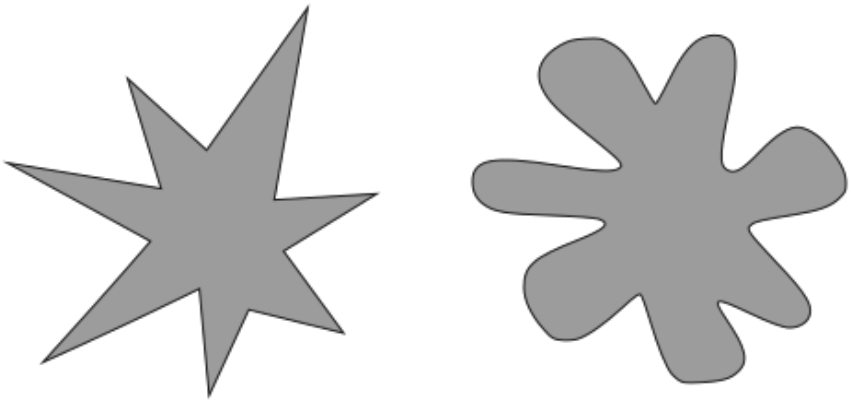

Figure 3.3: Kiki Bouba effect demonstration

Only one out of ten people would call the star-shaped symbol Bouba, the rest would call it Kiki. This is proof that our brain actually associates different shapes with different sounds.

Another Example of Ideasthesia could be hearing a voice in your head when reading. Our brain perceives visual information, and words, and transforms them into audio information, resulting in a reading voice in our head.

The audio and the visual cortices are both located at different places in the brain. The visual cortex is located in the occipital lobe at the back of the brain, whilst the audio cortex is located in the temporal lobe, or more specifically, the superior temporal gyrus. Yet we can see them interact. In the studies cited, we can read, that when we process visual information, mainly the visual cortex responds but we can also detect a response in the auditory cortex and vice versa.

In a new experiment relating to the Kiki-Bouba topic conducted by Emilio Gomez, they confirmed this by testing twice on groups of 100 and 50 people with ages varying from 10 to 80. They also asked different questions about the shapes or asked to name them differently for example ying-yang, tic-tac, or i-a all resulting in the same result, with the first of the two being the name of the first shape for 70% to 90% of the participants. This tells us that audio and visual cognition are definitely related.

Another important thing that relates to auditory and visual skills is response time (further RT). RT is the time it takes a person to react to, in this case, a visual or auditory stimulus. In this research on RT we can read how it is measured and described, the following paragraphs are based on this study[13]. In this study, they describe RT as follows:

„Reaction time (RT) is a measure of the quickness with which an organism responds to some sort of stimulus. RT is defined as the interval of time between the presentation of the stimulus and appearance of appropriate voluntary response in the subject.“

There are three described types of RT. The first is simple RT, where there is only one stimulus and one response. Then we have Recognition RT. Here there is a stimulus that should get a response and then another stimulus that should not. The last type is Choice

RT, where there are multiple stimuli and multiple responses. In humans, the response works thanks to having a nervous system. When the stimulus is recognized by the nervous system, neurons relay a message to the brain. After the brain processes the message it sends a signal to the spinal cord, which then sends a message to the part that performs the response, for example, our hand or fingers. It goes without saying that it is better for a person to have a quicker RT.

What can be learned from this experiment is, that visual RT gets a quicker response than audio RT. This can also show, that the two have to somehow relate to each other if the stimulus should receive the same response. For example, a person responding to an alarm clock that lights up and rings at the same time. If the auditory and visual RT would not be synchronized the person would either give two responses or be confused. It also shows another way to measure the audio-visual cognitive function.

3.2 How can we assess the audio-visual skill

There are several ways to assess the ability of audio-visual cognition. In some experiments or tasks, they are assessed separately and in some tests, the assessment is combined. In either case, audio and visual cognition are connected, so even with separate tests, in a way, both of them are assessed.

3.2.1 Multiple Objects Tracking

The following section is based on information found in this source [19] The multiple objects tracking test (further MOT) is a test of the ability to perceive and track more than one visual object at once. In the test, multiple objects are presented to the candidate. This could be viewed as the first part of the test. Next, some objects are highlighted. This tells the subject which of these objects they should track. Then the objects start moving. The subject's task is to track the highlighted objects. After all the objects move, the candidate is asked to determine, which are the objects he was supposed to track. The visualization of the MOT test can be seen in picture 3.4.

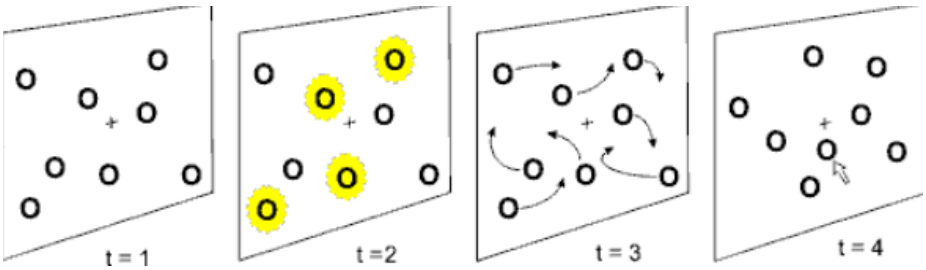

Figure 3.4: The MOT test visualisation

It is similar to the famous cup game. In the cup game, there are three cups. A ball is then placed under one of the cups. The cups are then mixed and in the end, the player is asked to identify the cup with the ball underneath (in this case we assume that the person mixing the cups is not rigging the game). In this scenario, the subject is tracking just one object, the other ones are there to distract him.

This test assesses the visual ability of a person to perceive multiple objects at a time. A person has limitations on how many objects they can track. What is more, the tracking

of objects can make the tested candidate totally unaware of different aspects of the objects, for example, their color. They can even be totally unaware of different visual stimuli. This was documented in the invisible gorilla test. If you want to try the referenced test, proceed before reading on [32].

In this test, multiple students are throwing around a ball. You are asked to count how many times the players wearing white pass the ball. At the end of the video, you are asked if you saw the gorilla walk through. The results of the experiment were that half of the people did not see the gorilla walk through the scene. Tracking more than one object can be very useful for example when trying to find some mistakes in a picture. It can also be mapped to being able to handle multiple tasks at once.

3.2.2 Test of attention in listening

The test of attention in listening (TAIL). The idea behind this test is to measure the RT of a person when a difference in audio stimulus is detected. The change can be for example a change in frequency or even the difference between silence and noise. In this test, the candidate is also subjected to other kinds of stimuli to distract him. Similar to the MOT test we try and distract the candidate with different things to test how well he can focus on just the one.

The subject is first told, what he should listen for, this can be a change of tone or a beep. Next different stimuli start to appear, for example, the flashing of a light. After a few seconds, the sound starts. The sound has to be continuous to make sure the candidate will respond to it even if it does not catch his attention at first. When the candidate hears the sound, he can for example press a button to signal that he heard it. The RT is then measured between the beginning of the sound and the button press resulting in a time assessment.

3.2.3 The Dichotic listening test

The following section describes the Dichotic listening test based on this source [7]. The Dichotic test is a common test that assesses the auditory attention of a person. In the test, the person has to be wearing headphones. In one ear a message is received that the person has to repeat aloud. In the other ear, the person receives another audio stimulus. The task is then to ignore the message being sent into one ear and try and repeat the message played in the other one as correctly as possible. The first test was conducted by Donald Broadbent in the 1950's. A variation of this test can also be done without headphones. In this variation, the tested candidate is played an audio recording out loud. In the recording there is a message, but also some „noise“. The noise is supposed to try and distract the candidate. His task is again to repeat the message he hears in the recording as correctly as possible. It is similar to the MOT test only the tested function is audio and not visual. The difference between the TAIL test and the Dichotic test is, that it does not measure the audio RT, but the way the candidate can resolve audio recordings.

3.3 existing games assessing audio-visual skills

Assessing audio-visual skills with a game can be pretty easy, mainly because the tests that were described are already games in a way. For example, let us look at the MOT test. Multiple games have been found, that do exactly what is described in the test, for example,

this game [30]. This game consists of many circles in a closed space. For the beginner mode, two of those circles are smiley faces. When the game starts, the smiley faces turn into regular circles. Then all of the circles start to move around randomly. In the end, the task is to click the circles which the player thinks are the smiley faces. The game can be seen in figure 3.5.

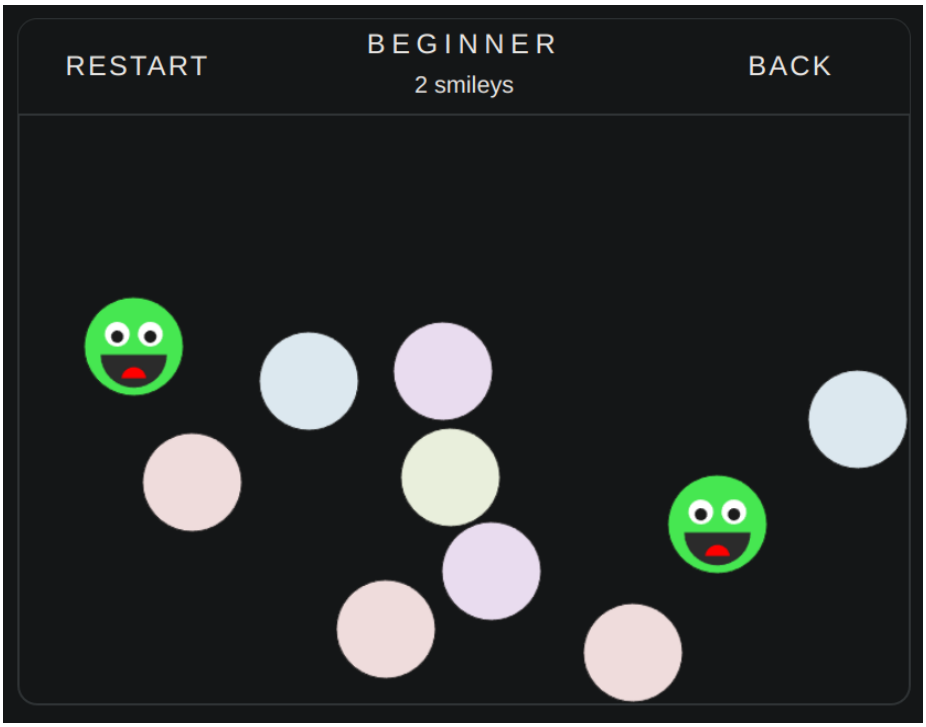

Figure 3.5: The MOT based game

This game is a literal depiction of the MOT test. The only way to test one's skill is by completing the five premade modes, beginner, intermediate, advanced, expert, and insane. This is a great idea for a game, only the testing part would need to change. There are a few ways to determine how good the player is. One way could be to design different game modes, as done in this game, and predetermine the assessment on some theory. For instance, the research showed how many objects a person can maximally track, that could be the insane level. Another way is to give the evaluation based on comparison to others. For example, if the game has 100 people playing it and storing their results. It could then be determined if the subject is better or worse than n% of the people, who have played the game. Obviously, the more people play the game, the better the assessment will be. Another pro of the comparing is, that the person can be compared to themselves and their increase or decline can be tracked in the specific skill.

Another game that was found, that is similar to the MOT test is the Squex mobile game (figure 3.6). The author of the game says that it tests one's reflexes. It actually tests the ability to perceive multiple objects like an MOT test. The game consists of 4 brown rectangles and a black square. As soon as the player touches the display in the bottom window he starts navigating the black square around the top window. Also, the brown rectangles start moving. The longer the player is playing the game the faster the rectangles get and upon collision the game ends. This is also a great way to evaluate the candidate

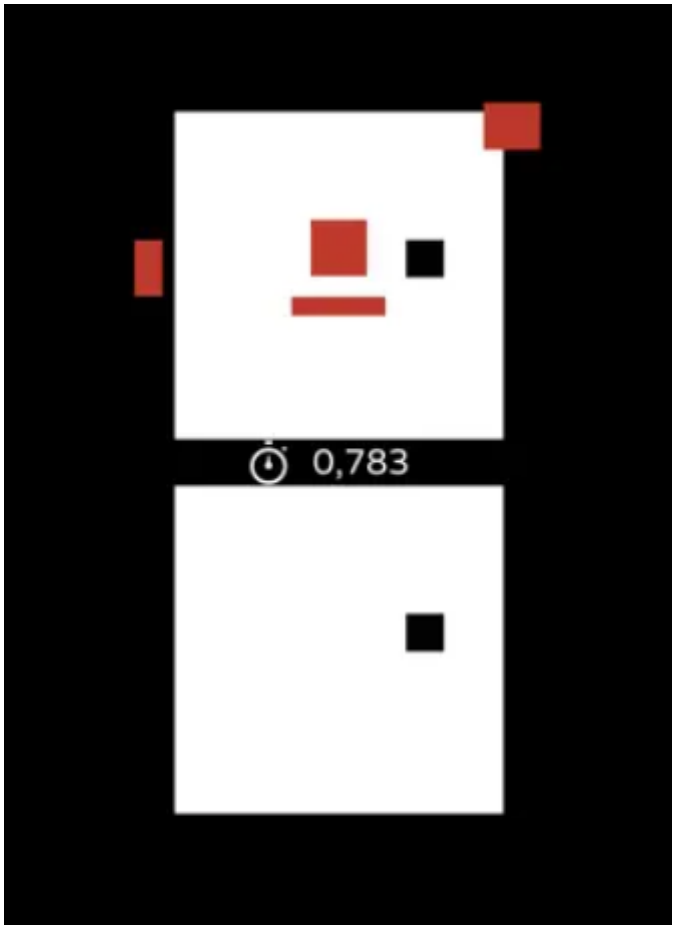

Figure 3.6: Squex game

with time. The time how long the player „survived“ completing the specific task can be measured, and then compared with others to evaluate.

The TAIL test can also be designed into a game pretty naturally, for example, this app/game, can test audio reaction time [31]. The app starts when you press a button. At some random time, the player will hear a noise, when he hears the noise, he has to press the button again. It then measures the time from the start of the sound to the button press just like a TAIL test. This game could also have levels similar to the MOT test game. In each level, some other stimulus could be added, like flashing objects, or even a different sound to distract the tested candidate. Again if some game-like features were added to this test, for example, achievements or a leaderboard (comparison to others) it can be turned into a game pretty easily. This game could be also designed in the same way only testing visual RT. Instead of hearing a sound to react to the player would for example wait for the screen or the button to change its color. After the change, the person would click the screen. In this game, we could also add harder levels, and distractions that could, but would not have to be from the same type of stimulus. For example, it could have the screen changing to a different color at a random time, and the candidate's task would be to wait for green, or it could add some audio distractions like randomly adding noises.

Designing games that assess audio-visual cognitive function seems like a more straightforward task. A lot of these games have been designed already a comparison of some of them can be seen on table 3.7. What would need change is the way they assess one's audio-visual ability. It can be seen that the main goal of the authors of these games was not the correct assessment of these cognitive functions.

Games	Squex	MOT-faces game	RT test game
which feels the most like a game with 3 being the highest	2	3	1
how the assessment is made	comparison to yourself	difficulty levels	comparison to yourself
test the game is based on	MOT	MOT	TAIL
progress logging	no	no	yes
one game	yes	yes	no
multiple minigames	no	no	yes

Figure 3.7: Comparison of games that assess audio-visual skills

3.4 Conclusion

It can be confirmed that the audio-visual cognition skills are connected to one another in a very strong way. In a way, the two cannot be divided, because the research shows, that when a stimulus is given, that should be registered by only one, it is registered by both. The neurons also send signals to the part of the brain where the other cognitive skill is processed. They are connected even though they are processed in different parts of the brain. This has been proven by many experiments, for example, the Kiki-Bouba effect. They show that what a person sees is also connected to some specific sounds and vice versa.

The connection can also be proved with calculating the difference of the speed of sound and the speed of light. The two are very different, which means, that some synchronization happens between them in the human brain. This also means, that when we are testing one of the qualities, we are also testing the other one.

A lot of tests can be done to assess one's ability of this specific cognition. A few of them are connected to RT. RT tells us how fast we can register a specific stimulus. It is measured as the quickness or time between the appearance of the stimulus and the adequate response. The shorter the RT is, the better you theoretically are in the given task. These tests then make a person do a specific task and measure his RT. What can also be done is measuring these qualities in a different way, for example, the MOT test. This test measures one's ability to track multiple objects at the same time.

Some of these tests can also be connected to memory. As mentioned in the reasoning part, memory is an important part of reasoning and when a person is doing any sort of test, even a test that assesses their audio-visual qualities, they will be conducting some reasoning in the end. That means memory tests revolving around noise can also be very

helpful. Another way to measure these qualities is attention. This can be seen for example on the Dichotic test or the TAIL test. These tests try to distract the tested candidate with different stimuli and then log how that affects his response or RT. This is also the case for the MOT test where every other object, that the candidate is not supposed to track is basically a distraction. Some games/tests that assess these skills already exist.

What has been found during the testing of these games is, that they either feel more like tests than games, or the assessment is not done very thoroughly. As described, making the test feel more like a game can have positive effects on the tested candidate. What has also been found is, that when doing these tests, the longer they take, the harder it is to pay attention resulting in worse results. The ones that feel more like a test than a game cannot even be described as games, even though it would be quite easy to change that. On the other hand, the games that really felt like games did not have good assessment qualities. Mainly because they were probably designed like games and not a way of assessment.

Assessing cognitive qualities can be done in many ways. One of them is defining the thresholds, or levels with theory and previous experiments. For example, in the MOT test, we could say that tracking more than eight objects at once is almost impossible, so who can do eight is on the highest level (the number eight is chosen randomly to describe the context and has no research behind it). Another way that one's cognitive ability can be assessed is by comparing them to others. If there is a game, where the player competes with users that previously played the game, he can then easily get a pretty accurate assessment based on the percentage. For example saying, that the candidate is better than 80% people that had previously played the game. This way of assessing also gives the candidate a more reasonable output. If levels are just designed based on theory, the result does not have to be that useful information. An interesting idea is the intersection of these two. The levels could be designed based on people playing it. With this way of assessing if, for example 100 people, played a prototype game (the more the better), the levels in the actual game could then be designed based on this experiment. This means the levels could indeed tell the player the wanted information, for example: „Only 80% of people were able to finish this level".

The goal is to find some common ground between the two designs, the one resembling a test, and the one resembling a game. Design a game, that feels like a game for the user, not like a test but also assesses his cognitive qualities as well as possible.

Chapter 4

Proposed Game design

This part talks about how the game has been designed. It summarizes why the game is designed the way it is, why it is believed to be perceived as a game and not a test, and also how the game assesses one's abilities.

While writing the theoretical part and doing the research it was realized that creating one game for assessing audio-visual and reasoning qualities is like chasing a mirage. It is a great idea at first, but after doing the theoretical research the conclusion that it could not be done properly came to light. The reason is that it is impossible to assess both of these qualities at once without the assessment losing credibility and accuracy. That is why it was decided to create two games as parts of one whole game application.

Generally speaking, it has to be noted that coming up with the reasoning game design was harder than the audio-visual game. The audio-visual game is quite natural because it is easier to assess the cognitive functions. It is defined how to assess them, and there is a finite amount of ways to do so. In the chapter before the problem of how to assess audio-visual games is described with many examples of tests that exactly tell us how audio/visual cognitive function can be assessed. There is the MOT test the TAIL test and also an assessment based on reaction times. Reasoning is a bit harder. The reason is that there are a lot of ways to define reasoning. Reasoning overall has a lot of different aspects to it. In reasoning an equivalent of reaction time does not exist because reasoning has so many different compounds. At first, there was the idea to implement a game that tests a few of these compounds (not all of them). The reason is that a game that assesses reasoning as a whole cannot be made because if we try to create a game that assesses all the separate compounds, the results would not be accurate. That is why the design focuses on just some aspects of reasoning and why the design is not a scenario-based game as defined in the chapter before.

4.1 Technologies used

4.1.1 Backend functionality

There are a lot of options to choose from when it comes to mobile game development. A lot of different engines, languages, IDE's. A few of them have been looked into before deciding which will be the best match for this project.

Godot

First, let us talk about Godot. Godot is a multi platform open source game engine under the MIT license. The main programming language used for programming games in Godot is GDScript. It is a custom made language specifically used in Godot. It has pretty lightweight syntax similar to python. Another option for coding in Godot is C#. The problem with C# and Godot is, that the integration is newer, and also the language was not specifically designed to work with Godot. This results in an irritating situation regarding sources and documentation. Even though C# is the more robust and overall well created language, most of the videos, documentation, code snippets and information regarding programming in Godot is in GDScript. For that reason most people use GDScript while developing in Godot. Overall Godot seemed as the most lightweight game engine from the ones that have been looked at. That does not mean you cannot build good games that have the potential to become very popular using it. For example the famous game Getting Over It With Bennett Foddy was built using Godot [27].

Unreal Engine

Unreal Engine is a game engine used mainly for developing 3D games. The language used for unreal engine development is C++ the reason being C++ is the fastest language out there which is well appreciated whilst developing 3D games. The game engine is developed by Epic Games. Its pricing is based on a royalty model with the threshold being one million dollars. This means that if the game revenue is more than one million dollars, Epic Games will take five percent of that. Unreal Engine seems like the most complex engine of the group. It displays great functionality and speed for creating 3D games. That is why most of the 3D complex and well known games are developed using Unreal Engine. Some examples include: Fortnite, ARK: Survival Evolved or Borderlands 3. These games are top of the industry games with Fortnite being the top grossing game as of the moment this thesis is being written.

For developing the game that complies with the design, Unreal Engine seems to be excessive and once the fact that the game will be 2D had been decided, Unreal Engine was not taken into account.

Unity

Unity is another multiplatform game engine. It is a proprietary software developed by Unity Technologies but it is free to use for noncommercial purposes. Using it for commercial purposes is also allowed until your game passes a certain revenue threshold being one hundred thousand dollars in twelve months. With Unity being a proprietary software it comes with more functionality than Godot. The Unity official documentation is not the best, but with more people using it there are bigger amounts of information on using the engine online than for Godot. The language used in Unity is C#. This is a great advantage compared to Godot, the reason being, that it is a well known programming language not used only in game development. According to these statistics [29], C# has been the eighth most used language in 2023 worldwide taking it's position right under Java and right above C++ and rising.

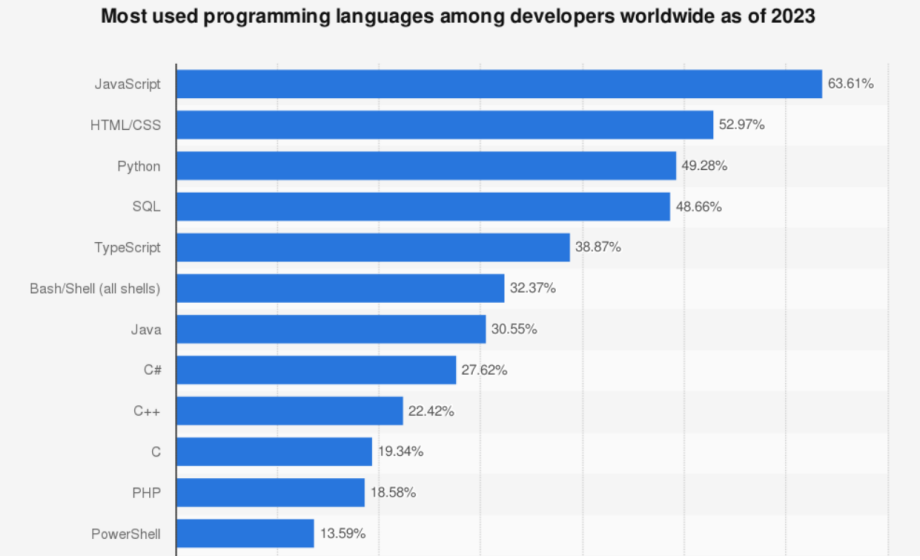

Figure 4.1: Survey showing most used programming languages of 2023 [22]

Unity lets the developer design mobile or PC games and it knows both 3D and 2D game development. It can also interact with Jetbrains Rider IDE for debugging the game whilst playing. Its UI toolkit is not so good, but you can get around that by using images that have been designed somewhere else for the sprites. A great thing is, that all the C# features can be used within the game. That means a database can be easily connected, which is crucial for this project. With C# being the favourable language, and with Unity being more robust than Godot, and not as complex as Unreal Engine it is the way to go for this project.

4.1.2 Database

A database is crucial for this project. It is needed for logging all the information of the users so the assessment can be done properly. We could just real-time assess the users qualities when he plays the game, but the assessment is better if we can track the information over time. It is important so that you can assess if the user is getting better or worse over time. With this feature, the game does not only assess how good the user is now, but it can show overall progress. With this method, it is also possible that the user will see he is getting better in these qualities while playing the game, since every time he plays the game he is basically training his cognitive functions. As of now that is just a theory, and it may or may not be proven during testing.

Relational Database

Relational databases are the most used and most known type of databases. Examples include MySQL, PostgreSQL, Oracle database and more. They are used in most cases for information systems and even games. The data is stored in tables with rows and columns where each row is represented with some unique ID. Its main advantage is that you can use it to mimic the model of complex systems. The database runs on some kind of separate server to which you send queries for retrieving the data. This approach is good if you have a lot of data. Another advantage of a traditional relational database is that you can have a

lot of different database objects, that are connected to each other using foreign ID's. This means, that the objects are all connected. In this case you do not have to explicitly store all the data you need in one table, but you can just point to the data you need from another table.

One interesting type of a relational database is SQLite. It works as a relational database, but it has no development server, the data is stored on the devices hard drives. This is a lightweight approach that can be very handy for storing smaller amounts of data. A big disadvantage is that you cannot store data from different users in one place. When coming up with the game designs one of the ideas was to use a SQLite database. It was then dismissed because the opportunity to get an overall assessment of how you compare to other users would be lost. In the end this option was not been chosen. What is needed is to store one type of object containing the reaction times and all other valid data every time the user plays the game, there is no need for any complex objects and relations between them. Eventually, another option was chosen for the database.

Firebase

Firebase is a type of no SQL database. The data is stored in JSON format and Firebase synchronizes this data for all the users retrieving the data. Firebase is the database technology that was chosen in the end. The fact that the application is going to need some type of user management, which is also a part of Firebase, was one of the reasons Firebase was chosen. This meant everything could be set up in one spot.

With only needing one type of object for the database containing the stats after every game played, Firebase real-time database seems like the best option. It is able to create and login users, and also store their data in a quite straightforward manner. The Firebase documentation for including it into Unity is also very well written, so there have been no problems integrating it into the game.

Another reason why Firebase seems like a appropriate option is the Firebase console. In this web application, you are able to see and manage all aspects of the database. This comes in particularly handy during the development and testing. You are able to delete users with the click of a button and the same goes for the database entries. When some residual testing data stays logged, you can just easily delete it, or even decide to delete all the data from one user specifically.

Firebase can also maintain a session meaning the state of a logged user is persistent throughout the whole game. It creates its own user ID which is also useful. Firebase also seems very safe user-wise. For example, even the administrator is not able to get to user passwords, this is a great feature. What it also does very well is retrieving error messages, which means that when there is an error during signing up or logging in the application can just display the specific message received from Firebase.

It has been decided to use an email and password type of authentication but Firebase has a large variety of options, for example authentication through Google, Facebook, Github, phone number and more. Another feature of Firebase is analytics, you can easily see how many users are using the application and how much traffic you get.

With the fact that no complex objects and specific relations between them will be needed for the project. Firebase seems like the best option.

4.2 Proposed Game Design

The goal was to create something that definitely feels like a game more than anything else but also assesses the subject's qualities precisely. The idea for the game after finishing the theoretical part was very broad at first. For the audio-visual game, the goal was to somehow connect the separate tests that were discovered during the research. For the reasoning game, the problem was how to figure out a way where the user can be given some information beforehand. The reason is that working with the information we receive, to come to a conclusion, is a part of reasoning. The question is, how can that be done without making the game seem like a test, without making it boring? The idea for the visual design was, to use basic shapes as a theme for the game. This way the game would have a nice clean look. Another reason for that was, the goal to make the user focus on one thing whilst playing the game, and not have a lot of distractions disguised as interesting design elements.

Three basic shapes are used to interconnect both of the games. The shapes are a triangle, a circle, and a square. The colors for designing the application are also important as can be read in this article about the effects of colors [15]. Specific colors have different effects on a person. For example, red can lead to quicker reaction times. Some other colors can also impact one's performance or feelings. Using for example red color to get the fastest reaction time from a person would result in an inaccurate assessment. That's why the game uses a mix of different colors. The colors used aren't sharp so that the person does not get affected, also a wide range of colors is used. The colors were then chosen using a color palette generator [25]. For the overall design, the decision was to go with black and white, There is no need for the game to be shiny and exotic.

The app consists of six scenes.

- User authentication scene
- Menu scene
- Audio-visual game scene
- Reasoning game scene
- Help scene
- Statistics scene

4.2.1 User authentication scene

The game uses a database to remember all the data needed for assessing a specific user. It seemed only logical to have users so their specific data could be remembered. The scene consists of two input windows one for the email and another for the password of the user. It also contains three buttons. A sign-in button, a log-in button, and a play as guest button. The sign-in button creates a new user using the Firebase authentication system, while the log-in button logs an existing user into his account. After successfully finishing one of these two operations the user is transitioned to the main menu scene. If something wrong happens during the authentication, an error message is displayed containing the specific message received from Firebase. Some examples of error messages are that the email you are trying to sign in with is already in use, a wrong password or email error, or a wrong format of the email address error.

The third button is the „Play as guest“ button. After clicking it the user is also redirected to the main menu scene. The difference is that no user is logged in and the app runs in a restricted mode. When the game is in restricted mode, the user can still play both of the games. He will still be displayed some of his results in a pop-up window right after the game ends, but he will not be able to transition to the statistics scene. The user then is not able to view his overall long-term statistics. The statistics for a guest user are not logged at all. The reason being, that it would just clutter the database, and there would be no way to retrieve data for a specific guest. Another difference to when a user is logged in is that the guest is displayed a message about these restrictions at the top of the main menu scene.

The code for the Sign In Scene is all handled in the SignIn.cs script and it is written based on the Firebase database documentation [26]. The only code that is not based on the Firebase documentation in this script is the logic for the button pressing where the user is transfered to the main menu scene, here is an example. This code also sets the boolean identifying the user as a guest to true, so that the game runs in the restricted mode. The Sign in scene is displayed in figure 4.2.

```
public void PlayAsGuest()
{
    GameManager.isGuest = true;
    UnityEngine.SceneManagement.SceneManager.LoadScene("MenuScene");
}
```

4.2.2 Menu scene

The menu scene is where you get transitioned after logging into the game. It consists of five buttons and nothing else. The idea is for the game to keep its lightweight design. This scene serves as a gateway to all the other scenes in the game. The buttons it contains and their functions:

- Statistics - transfers the user to the statistics scene
- Audio Visual game - transfers the user to the audio-visual game
- Reasoning Game - transfers the user to the reasoning game
- Log Out - the user is logged out and transferred to the sign-in scene
- Quit - the game is closed

The scene also contains a message box that only appears if you are logged in as a guest. The window notifies the user that the application is running in restricted mode. In this case, the user can still play the games, he will still see a general assessment right after the game ends in the game end pop-up window, but the statistics will not be saved into the database. The user is also forbidden to enter the statistics scene as there would be no data to display for him. The menu scene as displayed for a guest user can be seen in figure 4.3.

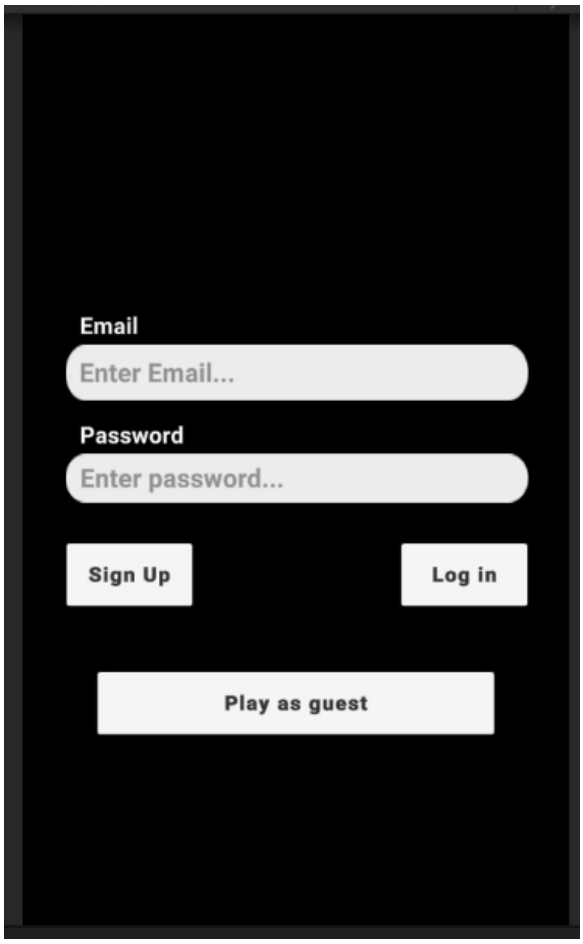

Figure 4.2: Sign In scene

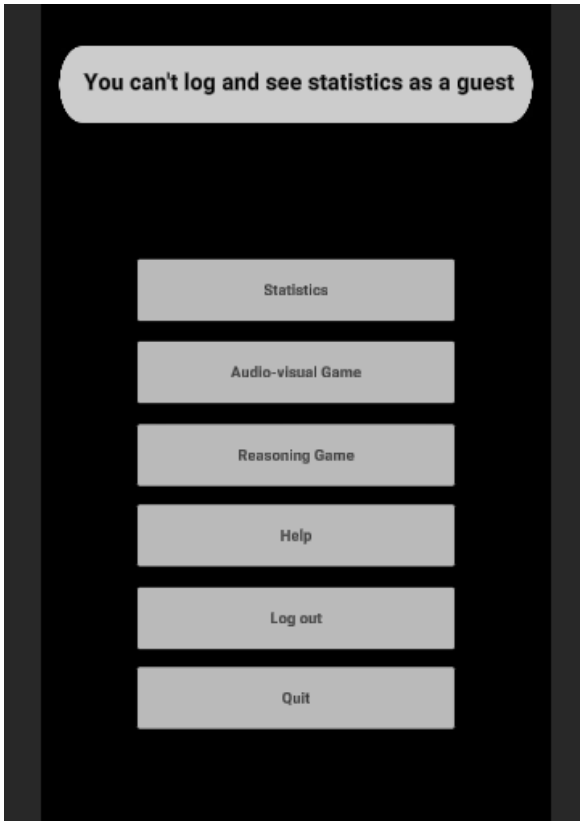

Figure 4.3: Menu scene

4.2.3 Audio-visual game scene

This scene contains the whole of the audio-visual game. The game itself will be better described in the Audio-visual game section. Here is just the description of the scene. Once we click the button audio-visual game button on the menu scene, the game automatically starts. Our options as a user are either to play the game or to use the back button of our phone to transfer back to the menu scene. The option to pause the game has been purposely left out. It is because this is an assessment game, and it feels like cheating, to pause the game. What the user could do in the pause is, for example, get ready to press some shapes. Having a pause button would result in people creating various tactics to get a better yet inaccurate result. The other option the user has is to play the game until the end. After that, a pop-up window appears that displays some overall assessment and points you to the statistics scene for more. The user also has the option to restart the game or transfer back to the menu scene with the use of two different buttons. The Picture displayed in figure 4.3 is how the scene looks in the Unity editor, you can see both the pop-up window and the game.

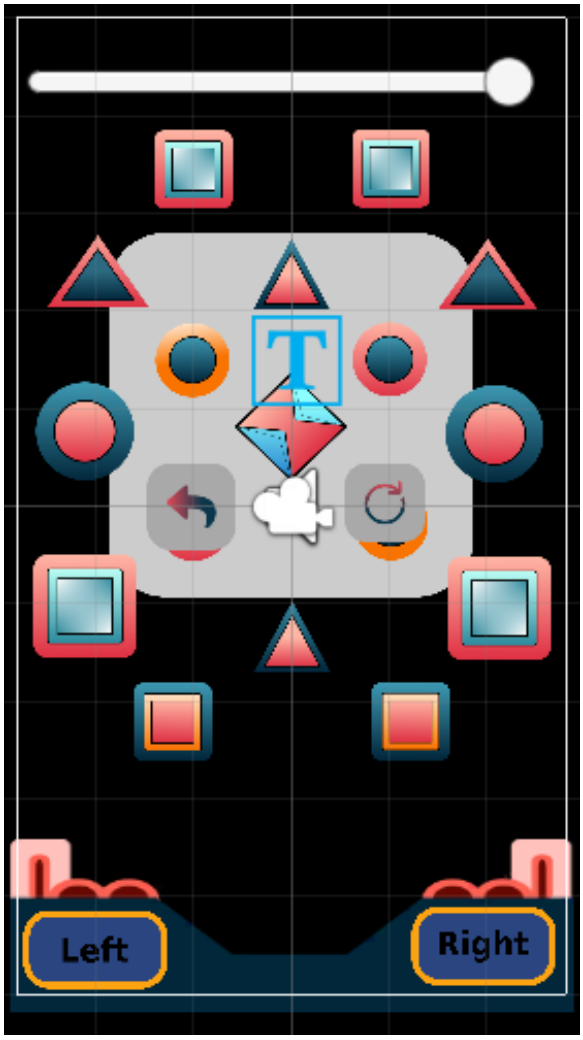

Figure 4.4: Audio-visual game scene

4.2.4 Reasoning game scene

The reasoning game scene contains the entirety of the reasoning game. Similarly to the audio-visual scene, you have two options as a user. One of them is to transfer back to the menu scene, and the other is to play the game and wait for the round end. The reason for this is the same as in the audio-visual game. To not let the user to be able to pause the game and think about his tactic. As mentioned in the chapters before, time is a crucial aspect when assessing reasoning and has an impact on the reasoning process. A difference between the audio-visual game scene and this scene is, that the player also has the option to go to the next level. There are ten levels to this game, so after the last level is played the player can only restart again or transfer back to the menu scene via the back button. The reasoning game scene is displayed in figure 4.5.

Figure 4.5: Reasoning game scene

4.2.5 The statistics scene

This scene may be the most important scene. Since this game is about the assessment of the user's cognitive functions, this is exactly what this scene does. The user gets a small assessment in the pop-up window right after the game ends, this scene shows the rest. The scene consists of several graphs that contain information about the user's cognitive qualities. At the top, the user also gets a comparison to others. Here he can read how many percent of people he beat. The main and most general statistics from each game are used for this comparison. The scene is scrollable because all the graphs would not fit just in one view. The user also has the option to press his mobile back button and transfer back to the menu scene.

This part describes the separate graphs. Since Unity does not contain any UI systems for displaying graphs, they had to be programmed separately. The programming of the graphs was inspired by this source [6]. Each graph consists of two axes, ticks on each axis, a text to describe each axis, points on the graph, and connections between them. There are two types of graphs, the first one is a graph that shows the audio-visual qualities, the y-axis on this graph starts from the number zero because you cannot get any negative values. The reasoning graphs have the x-axis in the middle of the y-axis, the reason is, that the value on the graph can also be negative. In these two types, the axes are drawn the same and the descriptions are also displayed statically.

Some parts are dynamically depicted. These parts include the ticks of each axis, the points, and the lines that connect the points. The ticks have to be displayed differently, because each graph, or most of the graphs has a different scale. For example, if the function measured is response time in milliseconds, the same ticks cannot be used for displaying the maximum count of objects at once. One of these axes has a range from 0-3000 and the other from 0-17. Another thing that is displayed dynamically is the points. As mentioned unity does not have any tool for this, what had to be done was to calculate the x and y value of each point. This also has to correspond to the specific scale that the graph has. The graph is also programmed so that you cannot run out of space. This means, that it calculates the distance on the x-axis between each point relative to how many points there are. The graph then uses its whole spread independently of how many points there are. If there are 2 points the whole graph will be used the same as if there are 4 points. The difference is, that the x distance between each point would be half on the x-axis in the first case and a quarter of the x-axis in the second case. The implementation is done this way, so that we can see our whole progress in one graph. The more points the user has, the more dots there will be, resulting in a curve that assesses the user's qualities the best.

Another thing that has to be programmed separately is the line connections between each dot. Each dot that is not in the middle or on the end has 2 lines coming out of it from each side. The lines are then connected together. This means that the length for each line has to be calculated. Then the calculation of the angles between all the dots is counted and the lines are rotated accordingly. The scene is displayed in figure 4.6.

Graph Descriptions

The first four graphs serve the same purpose just for different shapes. They display the reaction times separately for every shape. This means that the user can see the difference between his reaction time of the triangle, square, circle, and diamond. This statistic gives him an overall idea about his visual reaction time. It also shows if he prefers some shapes. The game starts with displaying just triangles, then the squares, and then the circles. The user can then see if that has an impact on the reaction time for specific shapes. Another thing he can measure is the reaction time for the diamond shape. This is an interesting statistic mainly because it can show that the user does or does not prefer the diamond if it appears. This statistic can show the user his reasoning quality regarding effectiveness. It shows whether or not he chooses the best shape for him, the best shape being the diamond because it diamond adds time, or if he just clicks shapes randomly.

The next, fifth graph shows the same statistic, but for audio. In theory, the audio response time should be higher than the visual. That is something the user can track.

The y-axis displays the same scale for all five graphs and that is time in milliseconds from 0 to 3000. The reason it goes from 0 to 3000 is that every shape stays on the screen for only three seconds, then it disappears. The same goes for audio, except the fact the time you have to react is two seconds. The reason for the user having more time for visual than for audio is that there are only two audio buttons yet there are seventeen different shapes that can appear.

The next graph is the „TimeLasted“ graph. It shows the overall time that the player used. This graph can be described as the most general description of the user's audio-visual cognitive function. It shows how long it lasted the player playing the game until his time ran out. The y-axis scale goes from 0 to 150, and it is one of the parts that can be

easily changed if needed. The time-lasted statistic is also used for calculating the user's comparison to others.

The seventh graph displayed shows the maximum number of objects the user can track. It shows how many shapes were maximally on the screen at once before the time ran out. This statistic is tied to the MOT test. It shows the user how many objects he is able to track at once. The y-axis scale goes from 0 to 17. The reason for that is that 17 different shapes can appear during the game.

The next graph is a graph that assesses the reasoning quality of the user. It shows how far off the user was from the desired amount. The x-axis is displayed in the middle of the y-axis because the y-values of this graph can be negative. The reason is that the user can either get a higher or a lower sum than the desired amount. This graph shows the user how close he is able to get to the desired amount. Another statistic that the user can see from the graph is how much he overshoots and how much he undershoots. What the user can take from this is how he handles risk. For example, if he often overshoots he does not mind risking and tries to get as close to the value without caring that he overshot. If the user undershoots often, he likes taking things safe. He fits in the desired amount and does not care that he does not get the best sum every time.

The next two graphs show statistics that have already been shown with a slight difference. They show the time-lasted for the audio-visual game and the final delta for the reasoning game. These are considered the main statistics, they are used to calculate the comparison to all other users. The difference is that these graphs show a different time frame. The points in these graphs do not show every attempt, but they calculate the average of a user for every day. This means that the user can see his assessment on a different scale. It can be useful for a more general assessment. The user sees his change over days and not attempts making the scale greater. The top of the statistics scene can be seen in figure 4.6.

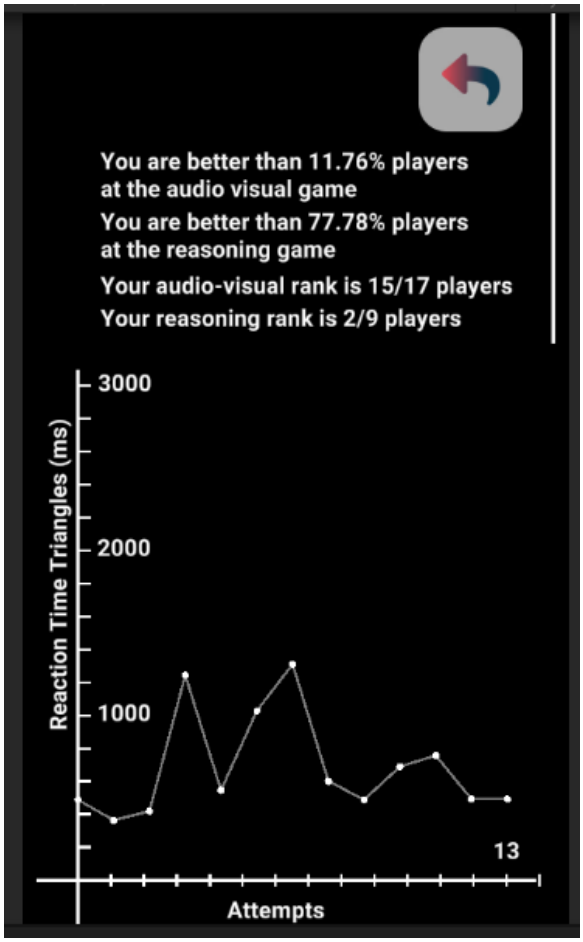

Figure 4.6: Statistics scene

4.2.6 The help scene

The help scene is a scene that contains instructions on how to play the game. The user can read the rules and conditions for both games. The user can go back to the menu scene by using the back button on his mobile phone. The help scene can be seen on figure 4.7.

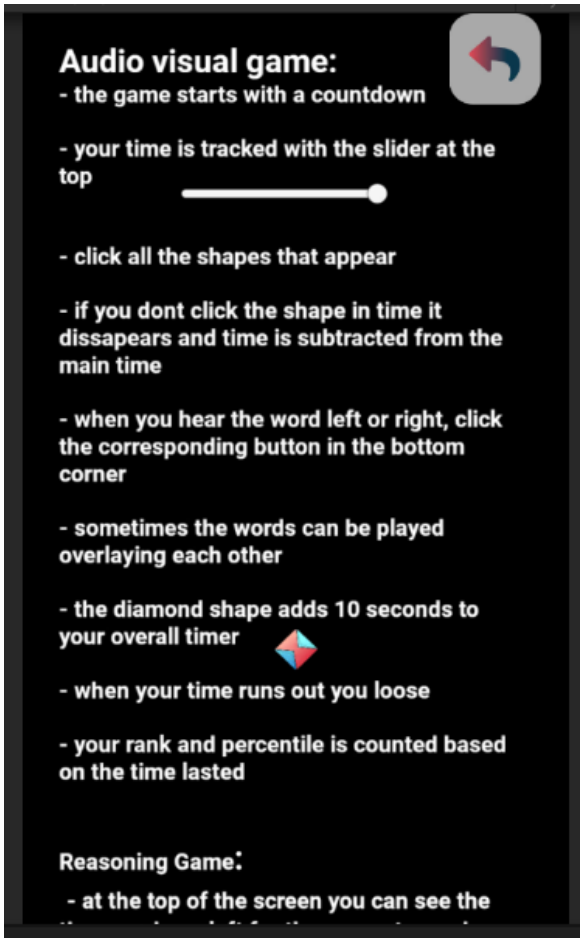

Figure 4.7: Help scene

4.3 Audio Visual game

The audio-visual game is used for measuring mainly the audio-visual cognitive functions of the user. It consists of seventeen shapes and two buttons. The game also contains a time bar at the top that indicates how much time the user has left.

The game starts with a countdown where nothing can be pressed. Then the shapes start to appear, first the triangles. There are four triangles in the scene. For the first twenty seconds of the game, only triangle shapes spawn. The user's goal is to click all the shapes before they disappear. The shapes stay visible for three seconds and this goes for all of the shapes.

Another event that also starts from the beginning with the triangles is the audio sources possibly playing. The audio sources play at a random time in the interval from 5-10 seconds. This means, that when the audio source plays, it cannot be played again for five seconds and also cannot be played later than ten seconds. This goes for both audio sources. There are two different audio sources, the voice saying „left“ and the voice saying „right“. When the audio source plays the user has to click the corresponding button in the bottom corner. When the user clicks any object in the scene a click sound is played to confirm the press was registered to the user.

Whenever the audio source plays the user has two seconds to react to the audio source. The reason for this is that the user does not click the left and right buttons randomly to optimize his final time. If the user reacts later than two seconds, time gets redacted from the main time.

After twenty seconds the squares start to appear. All together six squares can appear on the scene. They have the same properties as the triangle just a different time of appearance. The last shapes to appear are the circles. There are also six circles on the scene. The circles start to appear after the user is forty seconds into the game.

The reason for the shapes to start appearing at different times is so the game increases its intensity and is not the same from the beginning. Another way that helps this idea is the main timer. The time also gradually speeds up. This means the longer you play the game the quicker the time starts running out. This has been implemented so that the game has an endpoint.

The last shape on the scene is the diamond. The first time the diamond shape can appear is ten seconds into the game. The difference between the diamond and the other shapes is, that the diamond adds time to the timer. That is why it has different properties. Compared to other shapes it appears more seldom. The appearance interval is 10-15 seconds. This means that the fastest the diamond can appear is every 10 seconds and the slowest is every 15 seconds. When the user presses the diamond shape ten seconds are added to the main timer.

The overall goal of the user is to last as long as possible, the time that he lasted is also the main statistic used for assessing the user's audio-visual cognitive functions.

Here is a summary of how the game works in point form:

- The game countdown starts, and all the shape groups are visible
- Group by group the shapes disappear leaving none of the shapes on the screen
- Triangle and diamond shapes start to appear randomly. Audio recordings start to play randomly.
- After twenty seconds, square shapes start to appear.
- After forty seconds circle shapes start to appear.
- When the time runs out, the game ends displaying the evaluation window, and the statistics are logged.

4.4 Reasoning game

The reasoning game is designed to assess the reasoning cognitive function of the user. The scene consists of three different shapes, a countdown timer, the desired amount, and text that shows the user his potential sums. The game has ten levels that the user can go through before restarting, and it has a set time to finalize its decision at every level. The first level starts at twenty seconds. The time then gets gradually smaller with five seconds being the lowest it can go. The goal of the game is to select different shapes that each have different values to get as close to the desired sum as possible. The game is similar to the popular card game Blackjack. You try to get as close to the goal, but you do not want to overshoot. The goal in this case changes every time, and the variables are not the random cards, but the random number you will get from intervals displayed above the shapes.

At the top of the scene, you can see the countdown timer, under the timer there is a text that reads: „desired amount:“ and then the desired amount is displayed. The desired amount is a number that the user has to get to as close as possible. The best outcome is that the user gets the desired amount precisely correct. The desired amount gets higher at every level. It starts at the number 20 and is increased by 10 every round.

The game consists of 10 rounds, if the user wants his statistics to be logged, he has to finish all of them, because the game calculates an average result from all the rounds as the main statistic.

Beneath the desired amount the shapes are laid out. Each shape has a different potential value. The potential value is described above the shape with an interval. The player can use maximally eight shapes in each round. A potential value means, that the player does not know the actual value until the sum is counted at the end of the round. Let us say that the potential values of a triangle are 1-2. This means that if the player chooses two triangles, each of them can have a different amount ranging from 1 to 2. This results in four combinations. This also means, that in this case, there are three unique potential sums: 2, 3, and 4. The potential value interval of each shape gets increased every round. This results in a harder game state for the player. With the first interval being only two values big it is not so hard to get close to the answer. If the interval is for example ten values big, more combinations are resulting in more potential sums.

To prevent the game from becoming a counting game, at the bottom of the scene there are three statistics displayed. The user is told what is the minimal potential sum, the middle potential sum, and the maximum potential sum. This gives the user a general idea of where his guess will be. For example, if we have one of each shape clicked by the user and the intervals of the potential values are 1-4, 5-8, and 9-12, the minimal value the user can get is 15. That would happen if every potential value was the smallest number from the interval. The middle value would be 21 and the max value the user can obtain would be 24.

The game then assesses the user by showing him how far he is off from the desired amount as described below. The game also motivates the user to preferably stay under the desired amount and not overshoot by praising the user at the end of the level and telling him he can try again if he overshoots. Because of this, we can assess the user's risk management as described in the next section.

Here is a summary of how the game works in point form:

- The game time starts to tick and all of the three shapes are visible.
- The user can choose to select shapes from the screen by clicking them. He can also remove them using the minus button.
- When the time runs out, the final sum of the user is calculated.
- The user is informed about his result in the evaluation window.
- The user can then choose to restart the game, go to the next level, or transfer back to the main menu.
- When the user finishes all ten levels, the statistics are logged.

4.5 How the assessment is being done

This part of the game is the most important. The whole thesis is about assessing the subject's cognitive functions, not about creating the best and the most unique innovative game. With that being said, the goal also was to make the game fun to play, catchy, and unique.

How we can assess people's cognitive functions is described in the first two chapters. The goal here was to take that and hide it behind a facade of an engaging and fun game. The main premise used throughout the whole process of designing the game was, „Would someone play this in their free time, would someone play the game if they had ten free minutes on the bus?". That was the benchmark for making the game engaging, but the main goal here is, as said, how to assess the candidate.

What the game does is it logs specific information needed to assess the active user, what information and how it is logged is specified in the paragraph below. These logs are used to display graphs of the user's progression in the specific aspect. For example visual response, time and so on. The user can then see how well he does in specific fields of a cognitive function, and mainly he can see how he is progressing over time. This is crucial for the assessment. For a quality assessment, the user should not just get one result based on the game he just played, but he should be able to track all of his games and see if he is getting better or worse. Another very important thing is comparison to other users. Each game has a main statistic showing an overall assessment. For the audio-visual game, it is the time the user lasts playing the game, and for the reasoning game, it is how far he is from the desired amount. The application then counts an average from his logs and compares that to others. The game counts how many users are worse than the user and how many are better resulting in a percentile, which tells the user percent-wise how many users are better or worse than him. The game also shows the user his specific rank. This can give the user a feeling of how he compares to others. It also motivates the user to play as well as possible. This is also crucial for a good assessment.

4.5.1 How the assessment of audio-visual cognitive functions is being done

As mentioned before, assessing audio-visual qualities is easier because it is well-defined. What had to be done was taking the official tests and disguising them as a fun-to-play game. The assessment used in the audio-visual game is done by measuring response time to different stimuli, audio and visual. It also tracks how many shapes you can handle at once hence the MOT test. Another thing tested is how you respond to distorted audio. Whilst playing the game, we will notice, that sometimes the audio recordings overlap. That is not a mistake, it is done on purpose. It is important for the assessment to know if the person can handle overlapping audio as well.

For the visual cognitive function, the game logs various information, it logs the user's response times for various shapes. This means that the user can then see if his response time for one object is higher than his response time for other objects. This can be useful to assess if the user has a preference between shapes. It can show how the situation usually resolves if two or more shapes are spawned at once. The game only logs the best time for the user's response. The reason for that is, that logging every response time would make the data very cluttered and unreliable, this way, the best time for one attempt can be shown.

Also, the fastest time is the most important time. It is needed to assess what the user's capabilities are, and this can be done only with the fastest reaction times.

Another interesting thing is, that even though this game is designed as an audio-visual test, to some degree, we can see the candidate's reasoning qualities as well. As defined in the game rules, the diamond shape adds time to the user's overall time. With the response times, we can measure if, and how much the user prioritizes the diamond shape. If he does it shows that he is able to incorporate a reasoning aspect to the game and that he is not just pressing shapes randomly.

Another thing logged is audio response time, this again shows how fast the user responds to audio stimuli. In theory, it should show, that one´s response time to audio is higher than their response time to visual stimuli. As said before the audio loops can play at once, this feature was incorporated, so that the assessment is more general, and resembles the real world. In the real world, we get tons of audio stimuli at once all the time. That is why the game is made more like a Dichotic listening test. Even though the game logs only your best response, it is as similar to the real world as possible.

What is logged next is a variable called „maxObjectCount“. This information mimics the MOT test. By logging this, the game can tell the user how many objects he is able to track at once before his time runs out.

Every time a shape appears the game sends an event through the game manager to increment the overall active shapes count, and every time a shape disappears, with the user clicking it or with it showing for too long it decrements. It only rewrites the final maxObjectsCount if the number is higher than the one stored in the variable at the moment. This means it always returns the maximum number of shapes there were on the screen at once during the entirety of one game round.

How the game assesses overall audio-visual skill is by tracking the time the game lasts. This measurement does not track anything specific, it tracks how well the user does if all the aspects mentioned before are combined. That is why it is able to show some general assessment of his audio-visual cognitive skills to the user. The longer it lasts playing the game the better for the user. With this statistic, the user can see if he is overall getting better or worse playing the game. This is also the best way to assess how well the user compares to others. With the other measurements being very specific, this is the best way to do so. The number of players that have a worse time than the playing user is counted. This results in a percentile and a specific rank telling the user how well he compares to others. This is the best way to show the user some kind of overall assessment and comparison to others.

The object logged into the database is a simple class holding all the statistics specified. The object has to be marked as serializable. That is a keyword in C# that tells us the object can be serialized into a series of bytes and then reconstructed back. It is crucial to logging the object to the Firebase database because it needs to be sent over the network in a specific format. The object for the audio-visual game looks like this.

```
[Serializable]
public class DataToSave
{
    public string playerId;
    public float fastestReactionTimeSquares;
    public float fastestReactionTimeTriangles;
    public float fastestReactionTimeCircles;
    public float fastestReactionTimeDiamond;
```

```
    public float fastestReactionTimeAudio;
    public float timeLasted;
    public int maxObjectCount;
    public string shapeType;
    public string GameType = "AV";
}
```

4.5.2 The assessment of reasoning cognitive function

As mentioned, assessing reasoning is harder than assessing audio-visual cognitive skills. The reason is, that it is very hard to define reasoning. There are many different definitions, and that is why the actual cognitive function is so broad. There are not specific routines and tests to use for assessing reasoning. What can be done is trying to assess an overall general statistic, for example with a scenario-based game. Another option is to focus on one part of reasoning. One of the important aspects of reasoning is how a person can work with received information and data. That is another reason, why it is hard to assess with a game. The user has to be given some specific information beforehand, which can make the user perceive the game as a test.

As mentioned, the game consists of 10 rounds. For the assessment to be logged the user has to finish all out of the 10 rounds. It is because the rounds get increasingly harder. If the assessment would be based on just one round, the user could just play the first, easiest round, and then restart the game. This would result in him having a way better assessment than other users.

The game also logs various information from each round similar to the audio-visual game. It logs the desired amount which is the number that the candidate should get as close to as possible. It also logs the final amount that resulted from the sum after the round end, then it logs the level and the desiredFinalDelta. It also stores the user's overshooting coefficient, which tells the user how many times he overshot.

The desiredFinalDelta is the most important variable. This variable is used to assess the reasoning function of the user. It shows how far the user is from the desired amount. With this statistic, the game can track how often, and also how much the candidate under or overshoots. This can tell many things about the user. It can show if the user likes taking more risks or plays the game safely. The game focuses on this important aspect of reasoning. The game can also calculate how far the user is from the desired amount in absolute value. This statistic shows an overall reasoning assessment. It shows how far off the user usually is. This statistic is used to compare the user to other players because it is the most general reasoning assessment in the game.

The game also stores the overshooting coefficient. This is a variable that tells the user how many times he overshot out of all the 10 rounds. It generalizes the risk management assessment.

The application calculates players average of how far they are from the desired amount, it then compares this average with other users. This results in a percentile and a specific rank, where the user can see how many players, he beat. Again, this is crucial to a proper assessment. The user has to be compared to others to get a general idea of where he fits in the spectrum of all users.

Chapter 5

Implementation

The game was implemented in Unity using C# and Firebase for the database. All of the project scripts are stored in the Unity Assets/Scripts/ folder. The implementation is divided into four parts. The parts are the root part of the Scripts folder, the AVGame part, the ReasoningGame part, and the UserManagment part.

5.1 The root of the Scripts folder

The root part is the control room of the game. It contains all scripts handling game events. This part also handles the saving and reading from the database. The UserManagment part contains a script SignIn.cs. The script implements the user authentication logic. It was implemented using the Firebase Unity documentation [26].

Handling events is a very important part of the implementation. It is needed when one object wants to communicate with another. An event in C# is defined as so:

```
public static event UnityAction onShapeClicked;
```

An event is a specific type that can be called using the Invoke method. It has two functions, you can either invoke it or subscribe to it and listen for the invocation. The events are mainly used for communicating between objects that are defined in different parts of the code. That is why all the event handlers are defined in the root part. The specific event handler files are:

- GameManager.cs
- LogStatisticsEvents.cs
- MinMaxMidEvents.cs
- ObjectCountEvents.cs

They all handle different communication.

5.1.1 GameManager

The GameManager handles the main communication between objects. This means it handles click events like shape clicked or shape missed, restart game and next-level events, text changing events, and level-finished events. The game manager also contains a boolean called isGuest. This variable is used to store the game mode. If a user plays the game as a

guest, this variable is set to true. The variable is then used to check the game mode when pressing the statistics button on the menu scene, and also to ensure that the statistics of a guest aren't logged to the database. The guest state is checked as displayed in the following code snippet:

```
public void Statistics()
{
    if (GameManager.isGuest)
    {
        return;
    }
    UnityEngine.SceneManagement.SceneManager.LoadScene("Stats");
}
```

5.1.2 LogStatisticsEvents

This event handler is used for handling the information logging and retrieving. The game contains two serializable classes for logging statistics, one for the reasoning game and one for the audio-visual game. The classes are set within the scripts that handle the actual game processes. This event handler connects the game process scripts and the database logging scripts. This means, that when the game time runs out, and the corresponding class is set, it invokes the event, so the database logger knows that he can log the information. Another thing this handler does is, connect the statistics page with the data retriever. Because the data retrieving process is asynchronous, it could happen that the statistics frontend displays before the data is retrieved resulting in a problem. That's why events are used to wait for the data to be retrieved before the statistics scene is loaded. This is done by subscribing to these events. An example can look like this.

```
protected void Awake()
{
    LogStatisticsEvents.dataRetrievedTriangles += OnDataRetrieved;
    graphContainer = GetComponent<RectTransform>();
    InitializeTypeToListMap();
}

protected virtual void OnDataRetrieved()
{
    CreateAxis();
    CreateTicks();
    ShowNextGraph(type);
}
```

This snippet shows the subscription in the Awake() method and it also shows the OnDataRetrieved() method underneath. Each C# unity script can contain a built-in Awake() or Start() method or an Update() method. These are used as the main game methods. The Awake() method is called once the object is initialized. The Start() method is called when the object is first used, and the Update() method is an infinite loop that is called continuously while the object exists. Another built-in method needed for handling events is the OnDestroy() method. This method is called when the object is being destroyed. It is

important to unsubscribe to events in this method otherwise it could result in a problematic situation with the events subscription and also the object not being correctly destroyed. The code snippet below shows the destruction:

```
 protected void OnDestroy()
{
    LogStatisticsEvents.dataRetrievedTriangles -= OnDataRetrieved;
}
```

5.1.3 MinMaxMidEvents

This event handler is used for handling communication within the reasoning game. Every time a user clicks a shape in the reasoning game, an event in this file is called. Another function of UnityEvents is, that they can carry some payload data. In this example, it doesn't just invoke the event but it also sends an integer containing the present clicked count of triangles. This is an example of a definition of an event with a payload:

```
public static event UnityAction<int> sendClickedCountTriangle;
```

The event is listened to by a class that calculates the minimum, middle, and maximum value for the potential sum. The corresponding script then simply calculates the sum and Updates the displayed text.

5.1.4 ObjectCountEvents

This handler is used for keeping track of the amount of shapes on the screen whilst playing the audio-visual game. It contains two events. One event is for handling the shape appearance and one is for handling the shape disappearance. The events are listened to by the ObjectCountTracker.cs script. This script keeps track of the maximum shapes that were on the screen at once, and once the game ends it sends the info to the data sender script.

5.2 Audio Visual game part

This part is stored in a directory called AVGame. The root part of the directory contains the main scripts that implement the actual game. It also contains two other directories. One is called textUpdaters, and another one is called Stats. The textUpdaters directory contains a GameEnd.cs script. This script sets the evaluation text that appears on the scene after the game ends. It also hides the main canvas containing the game and shows the evaluation window.

The Stats folder holds all the scripts for displaying specific graphs. Each graph is implemented in its own script. This is done, because all of the graphs show different values, and some of the graphs have different scales. They also all wait for different data from the database. If the graphs were implemented in one script, they would be displayed iteratively. This way we can ensure, that all of the graphs are displayed at once because Unity builds all objects on the scene at the same time before the game starts.

5.2.1 Graphs implementation

The implementation of the graphs frontend was inspired by a programmer with an online pseudonym „code monkey“. The following paragraphs are based on information obtained from him [6]. The graph code initializes a private variable _dataGetter. This variable is an instance of the DataGetter class. There are two DataGetter classes, one for the audio-visual data, and one for the reasoning data. The application then uses this object to retrieve all the data needed for the specific graph. This is specified by the type of the data. The invocation is called in the Start method like so:

```
private void Start()
{
    _dataGetter.GetPlayerData(type);
}
```

The script then waits for the data to be retrieved as mentioned above. It then starts building the actual graph. First, it creates the axes. This is done by defining where the axes should be within the statistics scene. Then it uses the RectTransform component for drawing a line. Afterward, the game creates the ticks on the graph. The tick spacing is calculated based on the length of the axes, and the amount of ticks that should be displayed. Some of the graphs display more ticks than others, for example, the graph displaying the max objects count relating to the MOT test has 17 ticks, which is different, than the usual 15. The data on the x-axis is displayed dynamically. This means, that all the points fit into the x-axis no matter how many points there are. This means that the more the user plays the game, the more ticks there are resulting in a more accurate assessment. If the user plays the game only two times, the x-axis displays the number two at the last point. If the user plays the game three times it will display three and so on.

After this, the actual graph is painted. The data that the graph displays are stored in a dictionary. The dictionary consists of pairs of types string, and list. The string defines the type of values we want to use. It is assigned in the unity editor. The logged data is stored in the list, the dictionary for storing files looks like this:

```
protected void InitializeTypeToListMap()
{
    typeToListMap = new Dictionary<string, List<float>>
    {
        {"circles", _dataGetter.reactionTimeLists.reactionTimesCircles},
        {"squares", _dataGetter.reactionTimeLists.reactionTimesSquares},
        {"triangles", _dataGetter.reactionTimeLists.reactionTimesTriangles},
        {"diamonds", _dataGetter.reactionTimeLists.reactionTimesDiamonds},
        {"audio", _dataGetter.reactionTimeLists.reactionTimesAudio},
        {"timeLasted", _dataGetter.reactionTimeLists.timeLasted},
        {"maxObjectCount", _dataGetter.reactionTimeLists.maxObjectCount}
    };
}
```

The ShowNextGraph() method first stores all the values that will be needed from the dictionary based on the type. It then removes the default values. The default value is a value that is stored if it isn't rewritten. For example, if a square shape appears and the user doesn't click it in time, the default reaction time is 1000. The function then

calculates the last tick, this is used for displaying the description of the x-axis. Next, it calls the ShowGraph() method. This method calculates the position of each point. This is calculated based on the length of the x and y axis and on the scale of each axis. For all points, except the first one and last one a line is created that leads to the neighbouring points. This is done by calculating the distance between each point, it then calculates the angle between each point, so the lines rotate accordingly. This code is used for every graph. This includes the graphs for the reasoning assessment. The only things that change are the scales of the axes, the positions of the axes, and the data it uses to display them.

5.2.2 Root directory of the audio-visual game

This part contains the main logic for the gameplay. It contains 9 scripts. Each separate shape type has its own script, which means the triangle, square, circle, and diamond. The scripts are used for toggling the shapes appearance. When the game starts, all the shape groups (triangle, circle, square, and diamond) are visible. The game starts with a countdown, where the shape groups disappear group by group every second.

Each shape script handles the random appearance of each shape. It does this using Unity coroutines. A coroutine is a Unity method, that lets you run specific tasks asynchronously. This means, that once the coroutine is started, it runs in the background returning the control to Unity. Coroutines have the ability to pause execution and take control of the running code. This is ideal for handling the appearance of various shapes at once.

The script starts a coroutine that shows the shape after a random time within an interval. The user then has three seconds to respond to the shape appearance by clicking it. If he clicks the shape another Unity built-in method is called. This is the OnMouseDown() method. It is called whenever the user clicks the shape corresponding to the script. Suppose the shape is visible and clickable at the given moment. In that case, the script stops the active coroutine, plays the audio recording to ensure the user that he clicked the shape, toggles the visibility, and also invokes the event for a shape disappearing as mentioned before. What it does next is, it updates the reaction time and then starts a new coroutine to repeat the whole cycle. All shapes are implemented the same. The only changes are the time they start appearing. The game also measures reaction times for different shape groups, which is why they are all implemented in a different script. The script also listens to the avFinished event, so it knows when the game ends. When the game ends, the script sends the stored data to the script that handles data saving. Coroutines in Unity are defined using the IEnumerator class:

```
private IEnumerator StartAfterCountdown()
{
    yield return new WaitForSeconds(CountdownInterval);
    Clickable = true;
    RandomAppearance();
}
```

The root directory also contains the SliderTime.cs script. This script is used for controlling the game time. It maps a countdown timer to the Slider object. The Slider object is defined in the unity editor. Once the timer runs out, the script invokes the avFinished event, which notifies all other scripts that the time ran out. The game time is updated in the built-in Update() method. It decreases the timer value every tick of the game. The amount reduced from the main time gradually increases. This means the game gets infinitely harder over

time. This was implemented, so that the game always has an end. The time handling is implemented like this:

```
private void Update()
{
    if (timeRemaining > 0)
    {
        timeRemaining -= Time.deltaTime * _timeAccelerationCoeficient;
        _timeLasted += Time.deltaTime;
        timeSlider.value = timeRemaining;
    }
    else
    {
        TimeOut();
    }
    if(_timeAccelerationCoeficient < _maxTimeAcceleration)
    {
        _timeAccelerationCoeficient += 0.0001f;
    }
}
```

The root directory also contains a HandleNoise.cs script, that handles the noise the same way it does shapes. The difference is, that instead of a shape appearing it plays the audio recording and then tracks if the corresponding buttons were clicked.

The last two files this directory contains are the BackAV.cs file and the RestartGame.cs file. These files are used to invoke specific events when the corresponding buttons in the evaluation window are pressed.

The back button transfers the user to the menu scene, and the restart button calls an event that restarts all the data stored in the shape scripts and then reloads the audio-visual game scene.

5.3 Reasoning game part

The ReasoningGame directory also contains two folders. One of the folders also contains the logic for displaying graphs in the statistics scene. The scripts are implemented the same way as in the audio-visual game.

The ReasoningGame directory also contains a TextUpdaters directory. This directory contains all the scripts used for changing any text on the reasoning scene. Anytime some text field needs to be updated, an event is invoked. The text updater that is subscribed to the specific event invokes a method that updates the requested text. The events in the text updaters also send a string containing the new text as a payload of the event. Each text updater also contains a TextMeshProUGUI class object. This object is public and its corresponding frontend object is assigned through the Unity editor.

The ReasoningGame directory then contains three scripts, each corresponding to one of the shapes on the reasoning game screen. The main function of these scripts is to increase the intervals above the shapes. This is done differently for every corresponding shape. For the triangle shape, the only thing done is increasing the interval by two in each round. For the square interval, the script has to erase the bottom two values of its previous interval and then increase the interval. The interval has to be increased by four every time because we

subtract two values from it to start with. The subtracted values are added to the triangle interval. For the circle, the whole interval gets deleted, and a new one is created without the first four values from the interval before.

The shape scripts also handle the sum calculation at the end of the round. For every time the shape was clicked, the script generated a random number from the corresponding interval. It then sends the final sum through an event to the CountdownText.cs script. The random generation of the sum is implemented this way:

```
    private void GenerateSumWithRandomIntervalNumbers()
    {
        for (int i = 0; i < _clickedCount; i++)
        {
            int randomIndex = Random.Range(0, _potentialValues.Count - 1);
            _sum += _potentialValues[randomIndex];
        }
        GameManager.SendSumCircle(_sum);
    }
```

Other than these two functions the shape scripts communicate with the text updaters to update the corresponding UI elements every time a shape is added or subtracted. All of the scripts do the same job, the difference between them is how they handle the intervals, and also what event they invoke. The events the scripts invoke have the same function, but they all correspond to different shapes.

Another important script in the ReasoningGame folder is the CountdownText.cs script. This script holds the countdown timer. When the timer runs out, it invokes an event that tells all the shape objects that the game ended. The shape objects then send their final sums to the CountdownText.cs script so it can prepare the final evaluation. It calculates the final sum and sets the evaluation message. If the user overshot the desired amount, the message informs him about it. If the user didn't overshoot the message says „Good job!“. The logged statistics are then set. When the user finishes the last level, the average from all levels combined is calculated and sent to the data saver for reasoning. The class containing all the logged reasoning data looks like this:

```
    [Serializable]
public class DataToSaveReasoning
{
    public string playerId;
    public int desiredAmount;
    public int level;
    public int finalAmount;
    public int desiredFinalDelta;
    public string date;
    public int overshotCoefficient;
    public string GameType = "Reasoning";
}
```

The CountdownText.cs script also handles the overall game logic. It sets the levels based on the user choosing to go to the next level or to restart. It also updates the desired amount and the timer.

The last two files in the ReasoningGame folder are the same as in the AVGame folder. A RestartButton.cs and BackButton.cs file that either restart the game or transfer the user back to the menu scene. The folder also contains one script that we cannot find in the AVGame folder. This is the NextLevelButton.cs script that invokes the nextLevelClicked event telling all the scripts that the user chose to continue to the next level.

5.4 Database handling

The database is handled within four scripts.

- DataGetter
- DataSaver
- DataGetterReasoning
- DataSaverReasoning

Two of the files are used for retrieving data, and two for saving data. Each game has its own „saver“ and „getter“. The reason is that with this approach, reasoning and audio-visual statistics can be retrieved at the same time. The savers main method is called SaveData(). It is a method that is invoked by an event with a payload containing a class DataToSave. Every time the method is invoked it checks if all the data that is logged after every attempt has been stored into the DataToSave object. If not, the function returns. When all statistics have been added to the DataToSave object, the object is converted to JSON and saved to the Firebase database. This script also checks if the reaction time sent is the best reaction time for the specific shape. The database logs only the best reaction times. When the game ends, the SaveData() method is invoked many times, so it has to check, whether or not the time it just got is better than the one it has cached. The saving to the database was implemented using the Firebase documentation for Unity [26].

The DataGetter.cs script contains logic for retrieving data from the database. The main method in this script is called GetPlayerData. It retrieves data based on the type specified in the method call. This means it doesn't retrieve all data at once. With this approach the database doesn't have to retrieve all the data at once, but it can only retrieve the specific type it needs right now. The types include reaction times, the overall time lasted, and the maximum object count for the audio-visual game. For the reasoning game, the types are the level, the desired amount, the desired final delta (how far the user was off and in which direction), the overshot coefficient, and the final amount. Both of the data getters also retrieve the main statistic for the specific game (time lasted for the audio-visual game and the desired final delta for the reasoning game) grouped by days. This is further used to display the statistics on a different scale. The implementation of the data getters was done using the Firebase Unity documentation [26].

The data-getter scripts also contain logic for calculating the user percentile and rank. When the needed data is retrieved, the script calculates how many users are worse than you. This is then displayed as the rank. The amount of users below you is also used to count the percentile. The rank and percentile are displayed at the top of the statistics scene. If the user is logged in as a guest, none of the statistics get logged into the database.

The database structure is very simple. It has only two layers of data. The first layer is the stored user ID of the logged-in user. This ID is generated automatically by Firebase. Under each ID all of the users game attempt statistics are logged in the format of the classes

described above. This snippet shows the structure of the database for a user that had one attempt at the audio-visual game, and one attempt at the reasoning game:


```
{
  "users": {
    "3LwIH9z2Q6TrlxFG64sXtT27uKi1": {
      "-Nw4T8vNgd8Ctkhzrgwk": {
        "GameType": "AV",
        "date": "2024-04-22T13:02:22.5709670+02:00",
        "fastestReactionTimeAudio": 0.26667022705078125,
        "fastestReactionTimeCircles": 1000,
        "fastestReactionTimeDiamond": 1000,
        "fastestReactionTimeSquares": 1000,
        "fastestReactionTimeTriangles": 1000,
        "maxObjectCount": 4,
        "playerId": "3LwIH9z2Q6TrlxFG64sXtT27uKi1",
        "shapeType": "",
        "timeLasted": 14.018875122070312
      },
    "-NwM9klsNtunIqozMdPo": {
        "GameType": "Reasoning",
        "date": "2024-04-25T23:30:49.0513360+02:00",
        "desiredAmount": 110,
        "desiredFinalDelta": -2,
        "finalAmount": 106,
        "level": 10,
        "overshotCoefficient": -5,
        "playerId": "3cwmBfZRBAOF2wRRiMguI8DfH4j2"
      }
   }
 }
}
```


5.5 Dependencies and build

To open the Unity project there are some dependencies that need to be resolved. The implementation was done on Ubuntu 22.04 (Jammy Jellyfish) using Unity and JetBrains Rider for the programming and debugging IDE. The Unity version used for the project is 2022.3.20f1.git.6406910. The project was built with the Android settings for a 2D mobile game. When importing Android settings into Unity via the Unity hub, all major dependencies should install themselves. It is recommended to use the dependencies installed this way, using the Unity version specified above, for running the projects. This approach installs the JDK, Android SDK, Android NDK, and Gradle. Trying to run the Unity project with different versions is not recommended.

What is also needed for the project to run correctly is importing Firebase, this is very well documented in the Firebase Unity documentation. After logging in to the Firebase project, the Firebase SDK needs to be downloaded. Afterwards, the downloaded SDK has

to be unzipped into a desired location. The last part is importing Firebase into Unity. For this, click Assets > import package > import custom package. In the window that appears you can select all the packages that are needed for the project. The recommendation is to select all Firebase packages.

All dependencies should be resolved by following these instructions. If Unity sees any missing dependency, it will usually notify you, and help you install them straight through the editor.

To build the game successfully transfer the project files handed in with the thesis into a desired directory. After resolving all dependencies, the project can be opened using the Unity editor. Importing the user settings is optional, it will result in setting up the editor configuration the same way that was used during the implementation. For building the .apk files, click on File > build settings > build. Here it should be checked that all the game scenes are included at the top of the displayed window. By clicking build, the build process will start. When finished the .apk file will be stored in the desired location. The settings of the project as handed in are for the tablet version of the .apk file. If the resolution needs to be changed to mobile, all the main Canvas components in each scene have to be edited. When clicking the Canvas component, its settings will appear in the inspector window on the right. Here in the Canvas Scaler component, the Reference Resolution X has to be changed from 840 to 800. This will result in the resolution being transformed, so it fits mobile phones more precisely.

Chapter 6

User Testing

The main objective of the thesis is to assess users' cognitive functions. The best way to test this is to let users play the game and then ask them about their experience. The game assesses the user based on comparing his past results. It also compares the user to other users. The application is designed as a game. This means the user should not feel like he is being tested.

The testing does not just ask about the assessment part, it also rates how the application looks, and how well it is designed. This is a crucial thing for enhancing the user experience.

The game has two modes. You can play the game as a guest and also register an account. Guest users do not have access to the statistics page. Because of this, it is believed, that users who play as guests will have a worse rating of how well assessed they felt. This is because they only get a general assessment after the end of the game, but not a complex assessment as displayed on the statistics page. The user who plays as a guest also is not compared to other players.

Because the nature of this study is a pilot study, and because of licensing, the game was tested only on my family and friends. The potential testers were also limited because the application is created for Android phones (as defined in the assignment). The application was tested on 15 users. The users were not told any specific information on how to use the application before they played it. They were only told that it was a game that assessed their cognitive functions.

It is expected that the audio-visual game will be better received than the reasoning game. The reason is that the audio-visual game does not require thinking. People often link thinking to an activity that is hard or to an activity that is boring. Clicking shapes before they disappear is definitely easier and more relaxing than thinking of the best strategy for handling the reasoning game.

The game was developed using an Android phone. Specifically a Redmi Note 9 (128GB). The Android version on this device was 10 QP1A.190711.020. The game was also tested on a Lenovo and Samsung tablet. The Lenovo tablet was used for creating the bachelorTablet.apk file. This file is different. The application is built with different resolutions so it fits a bigger screen. The main development was done on the Redmi phone, this also means the main installer bachelorPhone.apk is meant for Android phones. The development phone displays with 4GB of RAM memory and the Octa=core Max 2.00GHz processor. The minimum Android version for the application to work is the Android 5.1 Lollipop. The android SDK used for development was the default version installed in Unity and it is the 25.2.0 version.

6.1 A summary of the testers

Most of the testers were Male (60%). Female testers took up 33.3% with one tester who preferred not to say. Most of the testers were aged 0-18 (73.3%) with the rest being aged 19-30. Realistically no users where aged 0, interval (0-18) was defined to keep the whole interval intact. The testers usually played the game more than once with the answer taking up 53.3% of the responses. 33.3% Of testers played the game just once. One of the testers (6.7%) played the game once every week before responding and one tester played the game a few times a week. 46.7% of the testers created an account with the rest playing as guests. The graphs corresponding to these questions can be seen on figures 5.1, 5.2, and 5.3

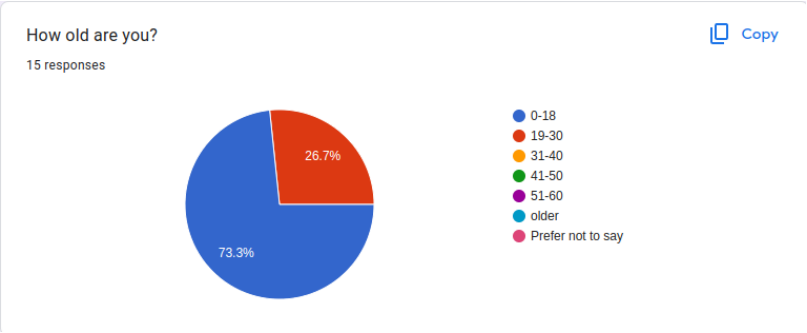

Figure 6.1: What is your age question.

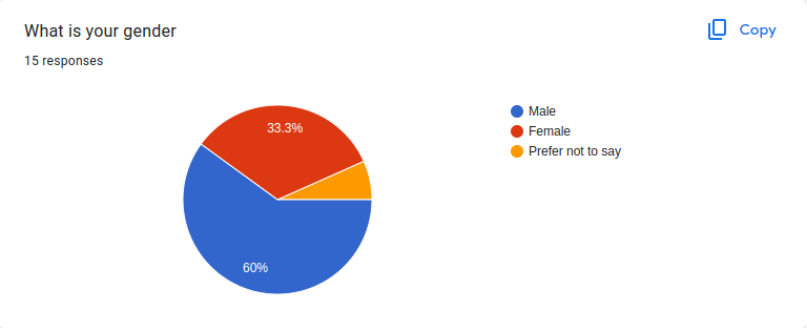

Figure 6.2: What is your gender?

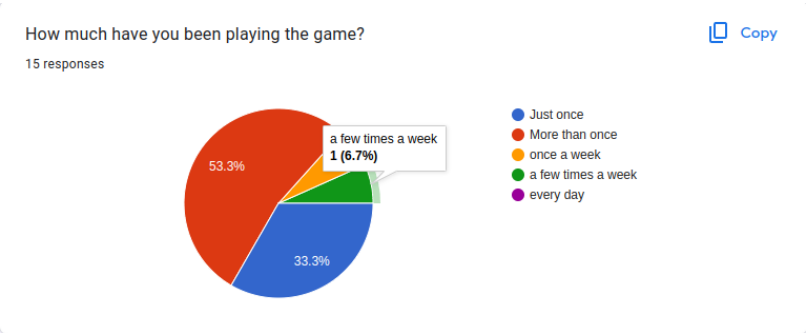

Figure 6.3: How much have you played the game?

6.2 Overall results

The average response for how the application was intuitive is 6.3 out of 10. This shows that the game has some room for improvement regarding UX/UI. With the application being a result of a pilot study and not really meant for production as it is, 6.3 is considered a good result. The overall look of the game was received a bit better than the intuitiveness with the average mark being 6.93 out of ten. Another question regarding the UI was how the testers rated the color scheme. The color palette chosen was rated at 4.13 out of five. This shows that testers generally liked the palette and did not find it invasive or distracting. The graphs corresponding to these questions can be seen on figures 5.4, 5.5, and 5.6.

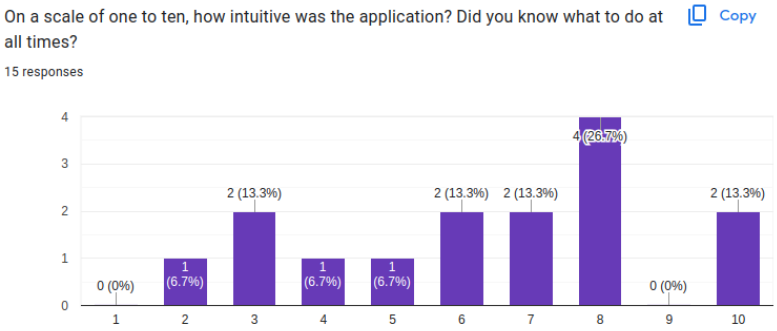

Figure 6.4: How intuitive was the application?

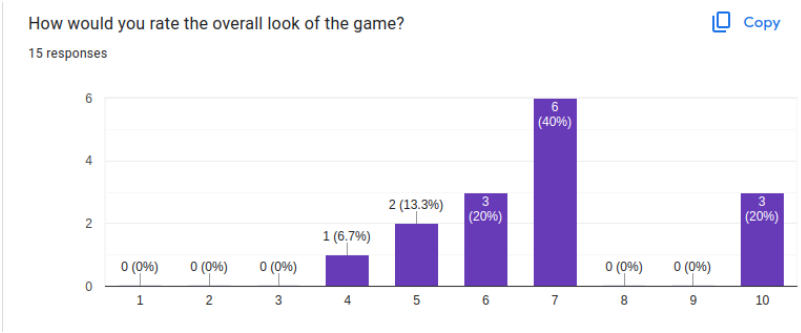

Figure 6.5: How would you rate the overall look of the application?

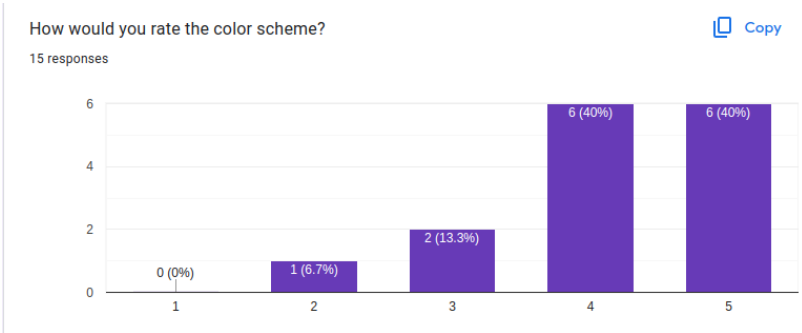

Figure 6.6: How would you rate the color palette?

The question: „Would you recommend the game to someone else?“ got a result of 3.4 out of 5. This is above the middle value. These statistics show us how enjoyable the games were. With the main goal of this thesis being the assessment of the users this is considered a good result. It means more than half of the people would recommend this assessment test disguised as a game to someone else. The graph corresponding to this question can be seen on figure 5.7.

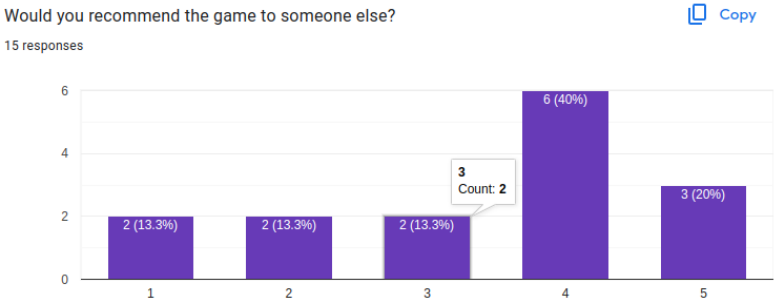

Figure 6.7: Would you recommend the game to someone else?

The audio-visual game came out as more enjoyable than the reasoning game with 60% of testers enjoying it more. 20% of the testers enjoyed the reasoning game more which leaves 20% of people that did not have a preference. These results confirmed the expectation. Some answers on why the tester enjoyed one game before the other include:

- „The audio-visual game was easier to understand.“
- „The audio-visual game was a bit easier than the other one“

This confirms that the audio-visual game is viewed as the easier more relaxing game. We can also see why a few of the respondents enjoyed the reasoning game more:

- „Because you need to think about it, and u just don´t tap like crazy guy.“
- „Audio was quite easy, so I pick reasoning“

We can see that these respondents did not conform to the premise that thinking is a boring or hard activity. The graph corresponding to this question can be seen on figure 5.8.

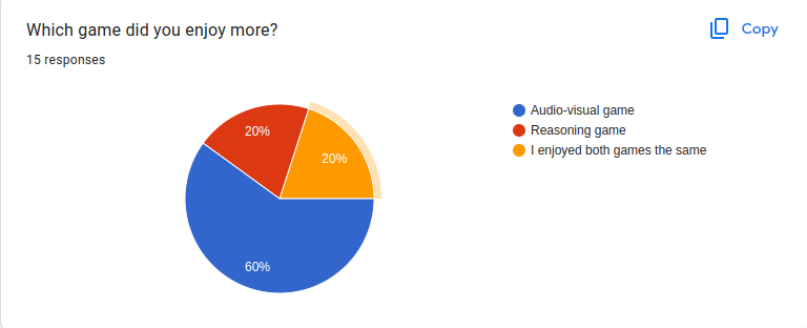

Figure 6.8: Which game did you enjoy more?

Some of the respondents also answered what part they specifically enjoyed about one of the games. Here are some of the results:

- „I enjoys the reveal, what sum i got.“
- „I enjoyed the detailed feedback“
- „Whilst playing the audio-visual game I enjoyed training my instincts and that I was able to see some progress after a while.“
- „The adrenaline from the time running out, it was basically fun“

In the second response, we can see that one of the users enjoyed the „detailed feedback“. This shows that the application can fulfill its goal at least for some users.

The next question is the most important question of them all. On a scale of one to ten, how well assessed did you feel? This question shows how the assessment is working according to the testers. The average result is 7.53 out of 10. In this question, we have to consider the game modes. The assessment is definitely worse when you play the game as a guest. In the responses, we can see, that the worst mark (5 out of 10) was chosen 5 times. All of these five times were by people who played the game as a guest. The worst mark registered by a tester with a created account is 7 and this was only one tester. The rest of the testers who created accounts responded mainly with the mark 10 (4 testers with created accounts). The rest of the testers with created accounts answered with the number 8. This question is the most important because it rates how well assessed the user feels. Assessing the user is the goal of the thesis. We can see that most of the users feel very well assessed, and the ones that do not play the game as a guest. This is considered a great result. The guest game mode is not designed to test the user extensively, it only gives a general assessment after playing a game. The user does not have access to the main assessment page. It is meant to show users what the game is about before they create an account. The corresponding graph can be seen on figure 5.9.

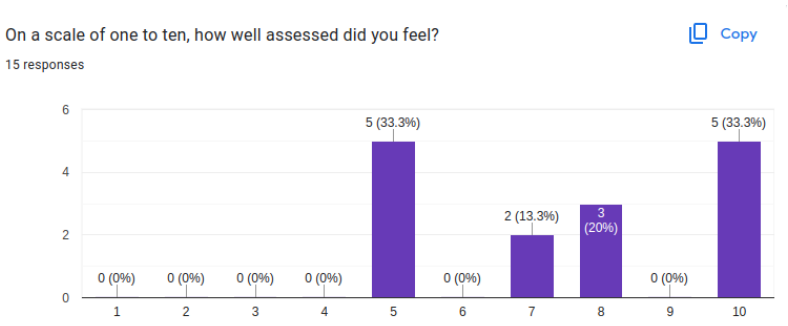

Figure 6.9: How well assessed did you feel?

For the question: „Is there any type of assessment you would add to the game?“ no one answered anything. This shows us that according to the users, there is not any assessment or statistic left out.

The next question asked if the comparison to others or the graphs was more important for the user. We can see that 46% of the users (7) said that the comparison to others is more important than the graphs showing the logged statistics. 26.7% equally found the graphs more important and did not have a preference. We can see that for the assessment the comparison to others is more important than the comparison to themselves. By comparing yourself to others you can get an idea of where you are regarding the whole population. The more people play the game the more accurate the result will be. The game shows you

your percentile and your rank amongst all the players. This feature is not just a way to compare yourself to others, but also a motivation for getting better and playing the game. The graph corresponding to this question can be seen on figure 5.10.

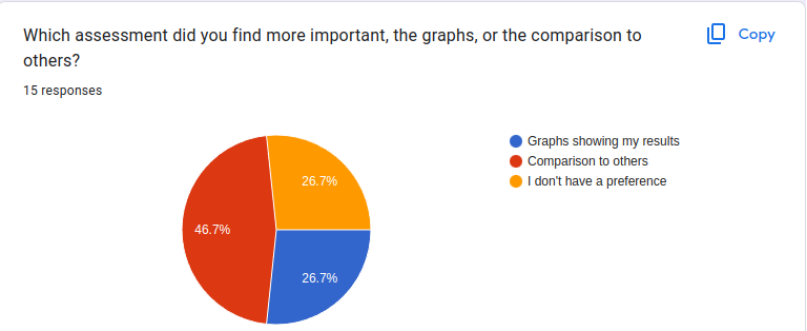

Figure 6.10: Which type of assessment did you find more important?

The next question tells us if the testers would play the game again with the average answer being 3.67 out of 5. The most common answer was 4 out of 5. No tester answered 0 which means never. The answers tell us that the game is overall considered engaging. The corresponding graph can be seen on figure 5.11.

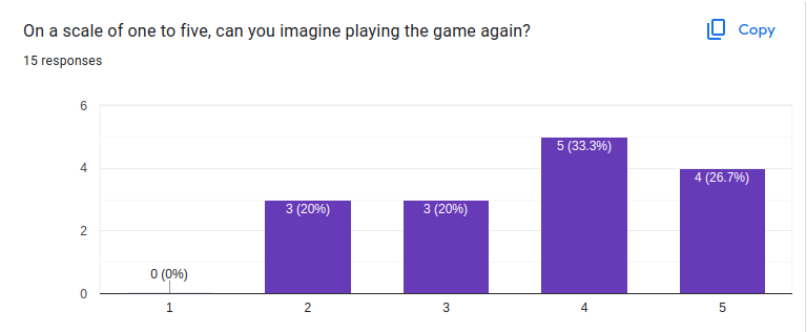

Figure 6.11: Would you play the game again?

Did the game feel more like an assessment or more like a game? This question has an average answer of 5.87. This is the lowest mark out of all the questions. We can see that most of the players answered that it feels like a game and not a test, but why is this the lowest number? Even though most of the players said, they would play the game again, or recommend it to a friend, a smaller half (7) of them said it feels like a test and not like a game. This could be because of the assessment features of the game. The application has a general assessment right after the game but mainly the statistics page that tells the user specific information about his qualities. This could be the reason for the game feeling like an assessment test for a big minority of the players. The corresponding question can be seen on figure 5.12.

The last question summarizes the overall experience with the application. The average answer here is 6.73. With this thesis being a pilot study this is considered a good result. The testers were mostly satisfied with the application yet there is room for improvement in the following studies. The corresponding graph can be seen on figure 5.13.

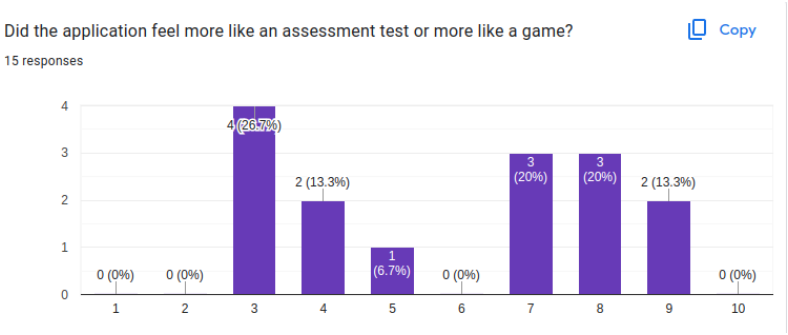

Figure 6.12: Did the game feel more like an assessment or more like a game?

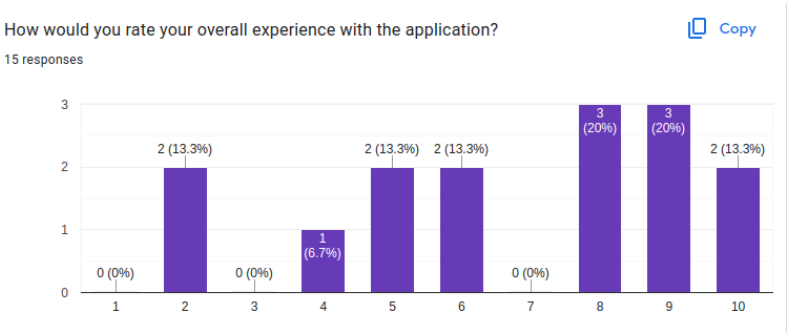

Figure 6.13: How would you rate your overall experience with the application?

6.3 Testing summary

All the testers used their own Android devices for installing and playing the game. The installation went through without any problems. From the questions we can see, that the game had a mostly positive rating. All of the answers were above the middle rating, with the worst rating being 5.87 out of 10. This question indicates that for a big minority of testers, the game felt a bit more like a test and not purely like a game. This begs the question, is this bad or not? Is it a problem that the users know they are being assessed? The users knowing that they are assessed could motivate them to pay more attention whilst playing the game. This would mean their assessment would be more accurate.

These results are considered very positive considering that this thesis is a pilot study. If the application would go into production, there would be room for some improvement. Overall the testers enjoyed the audio-visual game more. The reason for it is that users were more active, and it was not that hard to play. The reasoning game being hard is crucial for the assessment. If everyone were able to easily finish the game, the assessment would not make any sense, because the bar for assessing would be too low. The bar for the best assessment should be ideally set so high, that only one player, the best player can achieve it. With this setting, we would be able to distribute all the player skills in the assessment.

The assessment of the cognitive functions was considered high quality for all the users who created an account. The users who played as a guest rated the assessment out 5 out of 10. This result from the guest users was expected. Since the main goal was to assess users with a game, these results from the most important question positively support the outcome.

Chapter 7

Conclusion

7.1 Future Improvements

The final work does have some room for improvement. The main reason is that this is a pilot study and the application is not meant for production. One of the biggest things is the UX/UI design. Even though most of the respondents talked positively about both, it could be improved. Unity does not possess the best UI resources and is better for programming 3D games. It may be better to create a production version using a different game engine like Godot, or even an app framework not meant purposely for creating games, like Flutter. Maybe the best change would be just to create better PNG pictures considering all of the sprites are just vector graphics converted to PNG.

Another thing that could use some improvements is user handling. Right now users can create an account and log in, which is appropriate for this testing application. In production, it would definitely be good to add features like confirming the email address, different login methods, and the option to retrieve a password or delete an account. Another thing tied to this is the database rules. Firebase uses specific rules for handling what users can do. Right now the rules are set for testing, which means every user could in theory do whatever he wants if the application allows it. It would be good to have some database specialists look at this specific task. Another thing tied to the database is optimization. Since the database has not been tested on a lot of users at once, the behavior is unknown.

Another feature that should be considered is the option for the users to delete their data. This feature was thought of while implementing, but it was decided, that having the users delete their data would lead to less accurate assessment. The final conclusion if this feature should be added or not has not been made.

The option for the player to download his statistics in CSV format could be also added to the production version. The whole database could result in a very interesting dataset with more people playing the game. A feature that could be implemented in the future could be a module that predicts how your statistics will come out. It would be a machine learning module that would use the dataset of all players to predict their future outcome if they continue to play the game. This could in theory be useful for detecting illness specific to audio-visual or reasoning cognitive function.

7.2 Conclusion

The goal of the thesis was to create a game that assesses the user's cognitive functions, specifically the audio-visual and reasoning functions. The nature of this thesis is a pilot study. This means that extensive research had to be done in order to understand how these two cognitive functions work.

The research part of the thesis written in chapters two and three describes the two functions. It shows how cognitive functions work, and how we can assess them. The main reason for the research is to understand the cognitive functions due to its goal - implementing games that assess them.

The assessment part of each cognitive function shows different types of tests that can be used for assessing these functions. This part was crucial, as these tests are used in research for assessing these cognitive functions. This means that the tests could be used, they just had to be disguised as games.

Different games that are based on using these cognitive functions were analyzed. This showed that games implementing these tests do exist and that in the case of the audio-visual game, they are quite common. The games usually focused on just one specific part of the function and did not assess the user's qualities very well, or at all. The games that assessed the qualities well felt like tests and not games. That is why the goal was to create a game that feels like a game, not like a test. A game that the user would like to play in his free time. A game that he would recommend to others, yet it assesses his cognitive qualities as accurately as possible. This goal was fulfilled based on the user testing, where we can see that the average user answered 3.67 out of 5 that he would play the game again, and 3.4 out of 5 that he would recommend the game to other people. We can also see that the game was received as a game and not an assessment by most of the users with the average result being 5.87 out of 10. These results were considered positive due to the nature of the thesis being a pilot study. It is believed that in the future versions, these statistics could improve.

While concluding the research it was realized, that the final application would have to include two games and not just one. The reason is that making a game that assesses everything at once would lose its accuracy compared to assessing one of the functions at a time. With this fact, the application was implemented using C# and Unity. A Firebase database was also included for logging all the needed information.

The next part of the thesis is the game design. As mentioned earlier, the idea was to disguise the tests found during the research into games. The goal for the overall look of the application was to make it clean and minimalist. The reason behind this is, that it should be accessible to anyone. The scenario, where the user cannot play the game because he cannot navigate through the application needed to be minimalized. The audio-visual game connects four of the tests found during the research at once. It uses the MOT test, the reaction time test for both audio and visual cognitive functions, and the Dichotic listening test. It logs all the relevant information, so it can then display it to the user.

The reasoning game was harder to design. The reason is that reasoning is not a physical concept, but a psychological one. We can test audio-visual function with things like reaction time, MOT, and others. All of these tests measure a specific quality. With reasoning, this cannot be done. The game also had to give the user some information for concluding a decision because information received beforehand is a crucial aspect of the reasoning process. Even though it is harder to assess reasoning qualities, the game resulted in a fun-to-play and well-assessing game by most of the users.

After the application was created, it was tested on users. The overall results of the tests are quite promising. We can see that people mostly enjoyed the game. We can also see that they felt very well assessed which was the main goal of the thesis. Less than half of the users viewed the application more like a test than like a game.

It shows, that the application can have a real-world application for assessing cognitive functions of people. Users are able to test their cognitive skills. This could be useful for many reasons, for example, medical tests, or even testing in the hiring process. The results show that it is possible to create an accurate test for cognitive functions in the form of a game that people enjoy playing. This result could be better in future versions, but it will never be perfect due to the game needing to assess the user. When a user is assessed, on some level, he will always feel like he is being tested.

Bibliography

[1] Banissy, M. J., Jonas, C. and Kadosh, R. C. Synesthesia: an introduction. *Frontiers in Psychology*. 2014-12-15, vol. 5, [cit. 2023-12-20]. DOI: 10.3389/fpsyg.2014.01414. ISSN 1664-1078. Available at: http://journal.frontiersin.org/article/10.3389/fpsyg.2014.01414/abstract.

[2] Bloom, B. S. *Taxonomy Of Educational Objectives*. David McKay Company INC, 1956 [cit. 2023-12-09].

[3] Borter, N., Schlegel, K. and Troche, S. J. How Speededness of a Reasoning Test and the Complexity of Mental Speed Tasks Influence the Relation between Mental Speed and Reasoning Ability. *Journal of Intelligence*. 2023, vol. 11, no. 5, [cit. 2023-12-13]. DOI: 10.3390/jintelligence11050089. ISSN 2079-3200. Available at: https://www.mdpi.com/2079-3200/11/5/89.

[4] Capelle, J. D., Senker, K., Fries, S. and Grund, A. Deadlines make you productive, but what do they do to your motivation? Trajectories in quantity and quality of motivation and study activities among university students as exams approach. *Frontiers in Psychology*. 2023-12-5, vol. 14, [cit. 2023-12-13]. DOI: 10.3389/fpsyg.2023.1224533. ISSN 1664-1078. Available at: https://www.frontiersin.org/articles/10.3389/fpsyg.2023.1224533/full.

[5] Chen, Z. *Logical (deductive) reasoning*. Copyright © 2020 Elsevier Inc. All rights reserved, 2020. ISBN 978-0-12-816511-9.

[6] CodeMonkey. *Video for inspiring the implementation of graphs in unity*. [cit. 2023-04-1]. Available at: https://unitycodemonkey.com/video.php?v=CmU5-v-v1Qo.

[7] Conway, A. R. and Kane, M. J. 14 Capacity, Control and Conflict: An Individual Differences Perspective on Attentional Capture: An Individual Differences Perspective on Attentional Capture. In: Gibson, B. S. and Folk, C. L., ed. *Attraction, Distraction and Action - Multiple Perspectives on Attentional Capture*. North-Holland, 2001, vol. 133, p. 349–372. Advances in Psychology. DOI: https://doi.org/10.1016/S0166-4115(01)80016-9. ISBN 9780444506764. Available at: https://www.sciencedirect.com/science/article/pii/S0166411501800169.

[8] Dwyer, C. P., Hogan, M. J. and Stewart, I. An integrated critical thinking framework for the 21st century. *Thinking Skills and Creativity*. 2014, vol. 12, p. 43–52, [cit. 2023-12-11]. DOI: 10.1016/j.tsc.2013.12.004. ISSN 18711871. Available at: https://linkinghub.elsevier.com/retrieve/pii/S1871187114000030.

[9] El Baba RM, S. M. *Neuratomy, Frontal Cortex*. StatPearls, 2023. Available at: https://www.ncbi.nlm.nih.gov/books/NBK554483/.

[10] Gomez, E., Iborra, O., De Córdoba Serrano, M. J., Juárez Ramos, V., Artacho, M. et al. The Kiki-Bouba Effect A Case of Personification and Ideaesthesia. *Journal of Consciousness Studies*. january 2013, Volume 20, Numbers 1-2, 2013 , pp. 84-102(19), [cit. 2023-12-26].

[11] Gorilla, T. *Test Gorilla*. 2023 [cit. 2023-12-13]. Available at: https://www.testgorilla.com/.

[12] H., F. J. Metacognitive Aspects of Problem Solving. Hillsdale. 1976, [cit. 2023-12-11].

[13] Jain, A., Bansal, R., Kumar, A. and Singh, K. A comparative study of visual and auditory reaction times on the basis of gender and physical activity levels of medical first year students. *International Journal of Applied and Basic Medical Research*. 2015, vol. 5, no. 2, [cit. 2023-12-26]. DOI: 10.4103/2229-516X.157168. ISSN 2229-516X. Available at: https://journals.lww.com/10.4103/2229-516X.157168.

[14] Jung, C. G. *Collected works of C.g. Jung, volume 6*. Princeton, NJ: Princeton University Press, 1976-10. Collected Works of C.G. Jung. ISBN 0-691-01813-8.

[15] Kendra Cherry, M. Color Psychology: Does It Affect How You Feel? february 2024, [cit. 2023-04-10].

[16] Killgore, W. D. Effects of sleep deprivation on cognition. In: Kerkhof, G. A. and Dongen, H. P. van, ed. Elsevier, 2010, vol. 185, p. 105–129. Progress in Brain Research. DOI: 10.1016/B978-0-444-53702-7.00007-5. ISBN 9780444537027. Available at: https://linkinghub.elsevier.com/retrieve/pii/B9780444537027000075.

[17] Krathwohl, D. R. A Revision of Bloom's Taxonomy: An Overview. *Theory Into Practice*. 2002-11-01, vol. 41, [cit. 2023-12-13]. DOI: 10.1207/s15430421tip4104_2. ISSN 0040-5841. Available at: https://www.tandfonline.com/doi/full/10.1207/s15430421tip4104_2.

[18] Pierce, C. S. *Collected Papers of Charles Sanders Pierce*. Harvard University Press, 1960 [cit. 2023-12-09].

[19] Pylyshyn, Z. Multiple object tracking. *Scholarpedia*. 2007, vol. 2, no. 10. DOI: 10.4249/scholarpedia.3326. ISSN 1941-6016. Available at: http://www.scholarpedia.org/article/Multiple_object_tracking.

[20] Robert J. Marzano, J. S. K. *The New Taxonomy of Educational Objectives*. Corwin Press, 2007 [cit. 2023-12-11].

[21] Sadler, J. E. John Amos Comenius. *Britanica*. 24 Mar. 2024. Available at: https://www.britannica.com/biography/John-Amos-Comenius.

[22] Statista. Most used programming languages among developers worldwide as of 2023. *Statista*. 2023.

[23] Travers, R. M. W. *Research and Theory Related to Audiovisual Information Transmission. Revised Edition.* Utah Univ., Salt Lake City..Bureau of Educational Research., 1967.

[24] University, M. *Study shows how our brains sync hearing with vision.* 11. May 2021. Available at: www.sciencedaily.com/releases/2021/05/210511081149.htm.

[25] *Coolors.* [cit. 2023-04-10]. Available at: https://coolors.co/.

[26] *Firebase Documentation.* [cit. 2023-04-1]. Available at: https://firebase.google.com/docs/unity/setup.

[27] *Getting Over It With Bennett Foddy.* [cit. 2023-04-1]. Available at: https://store.steampowered.com/app/240720/Getting_Over_It_with_Bennett_Foddy/.

[28] *Mconsulting website describing different assessment based games for the hiring process.* [cit. 2023-12-18]. Available at: https://mconsultingprep.com/game-based-assessments.

[29] *Most used programming languages 2023.* [cit. 2023-04-1]. Available at: https://www.statista.com/statistics/793628/worldwide-developer-survey-most-used-languages/.

[30] *MOT game design.* [cit. 2023-12-21]. Available at: https://solvidini.github.io/multiple-object-tracking/.

[31] *Sound Reaction Test.* [cit. 2023-12-21]. Available at: https://play.google.com/store/apps/details?id=com.playbackfm.soundreactiontest&utm_source=global_co&utm_medium=prtnr&utm_content=Mar2515&utm_campaign=PartBadge&pcampaignid=MKT-Other-global-all-co-prtnr-py-PartBadge-Mar2515-1.

[32] *The invisible gorilla experiment website.* [cit. 2023-12-21]. Available at: http://www.theinvisiblegorilla.com/gorilla_experiment.html.

[33] *The McKinsey website.* [cit. 2023-12-19]. Available at: https://www.mckinsey.com/careers/mckinsey-digital-assessment.

[34] *The Talent Game service.* [cit. 2023-12-19]. Available at: https://thetalentgames.com/.