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 HUMAN MEMORY, ATTENTION AND REFLEXES

VÝVOJ HRY PRO HODNOCENÍ LIDSKÉ PAMĚTI, POZORNOSTI A REFLEXŮ

BACHELOR'S THESIS
BAKALÁŘSKÁ PRÁCE

AUTHOR DENYS PETROVSKYI
AUTOR PRÁCE

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

BRNO 2024

BRNO FACULTY
UNIVERSITY OF INFORMATION
OF TECHNOLOGY TECHNOLOGY

Bachelor's Thesis Assignment

153402

Institut:	Department of Computer Systems (DCSY)
Student:	**Petrovskyi Denys**
Programme:	Information Technology
Title:	**Game Development for Assessment of Human Memory, Attention and Reflexes**
Category:	Biocomputing
Academic year:	2023/24

Assignment:

1. Study and learn about human memory, attention & reflexes 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 human memory, attention & reflexes as well as limitations of the existing methods for their assessment.
4. Select and suggest changes in design of game (and/or digital brain test) for assessment of memory, attention & reflexes.
5. Implement the designed game (and/or digital brain test) for assessment of human memory, attention & reflexes.
6. Create a set of benchmark tasks to evaluate quality of the assessment of human memory, attention & reflexes.
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

This thesis details the development of an Android application designed to simultaneously assess memory, attention and reflexes. The main goal was to create an interactive platform that not only engages users in cognitive tasks, but also allows them to track their performance over time through statistical analysis and progress graphs. Using the Flutter framework, the application offers a user-friendly and clear design, integrated with Firebase for efficient user data management and authentication. The project involved the development of a series of cognitive modules integrated into a single game that dynamically adjusts to the user's results, ensuring accurate assessments and user engagement. The user interface was designed to be intuitive, ensuring ease of use and accessibility. User test results show that the application successfully measures cognitive abilities and provides users with valuable feedback, thereby supporting continuous cognitive improvement. This work demonstrates the potential of gamified cognitive assessments in mobile applications, contributing to the fields of educational technology and cognitive psychology.

Abstrakt

V této práci je podrobně popsán vývoj aplikace pro Android, která je určena k současnému hodnocení paměti, pozornosti a reflexů. Hlavním cílem bylo vytvořit interaktivní platformu, která uživatele nejen zapojí do kognitivních úkolů, ale také jim umožní sledovat jejich výkon v čase prostřednictvím statistické analýzy a grafů průběhu. Aplikace s využitím frameworku Flutter nabízí uživatelsky přívětivý a přehledný design a je integrována se službou Firebase pro efektivní správu uživatelských dat a ověřování. Projekt zahrnoval vývoj řady kognitivních modulů integrovaných do jediné hry, která se dynamicky přizpůsobuje výsledkům uživatele, čímž zajišťuje přesné hodnocení a zapojení uživatele. Uživatelské rozhraní bylo navrženo tak, aby bylo intuitivní a zajišťovalo snadné používání a přístupnost. Výsledky uživatelských testů ukazují, že aplikace úspěšně měří kognitivní schopnosti a poskytuje uživatelům cennou zpětnou vazbu, čímž podporuje neustálé zlepšování kognitivních schopností. Tato práce demonstruje potenciál herního hodnocení kognitivních schopností v mobilních aplikacích a přispívá k rozvoji oborů vzdělávacích technologií a kognitivní psychologie.

Keywords

game development, memory, attention, reflexes, cognitive assessment, flutter, firebase, cloud firestore, real-time analytics, user interface design, database management, cognitive functions

Klíčová slova

vývoj her, paměť, pozornost, reflexy, kognitivní hodnocení, flutter, firebase, cloud firestore, analýza v reálném čase, návrh uživatelského rozhraní, správa databáze, kognitivní funkce

Reference

PETROVSKYI, Denys. *Game Development for Assessment of Human Memory, Attention and Reflexes.* Brno, 2024. Bachelor's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor doc. Aamir Saeed Malik, Ph.D.

Rozšířený abstrakt

Rychlý rozvoj technologií a rostoucí nároky na každodenní kognitivní úkoly vyžadují vytvoření nástrojů pro hodnocení a zlepšování kognitivních funkcí, jako je paměť, pozornost a reflexy. Tato práce se zaměřuje na vývoj herní aplikace určené k současnému hodnocení těchto kognitivních schopností. Aplikace je navržena tak, aby poskytovala uživatelsky přívětivé rozhraní, které podporuje průběžnou interakci a usnadňuje sledování kognitivních trendů v čase.

Hlavním cílem této práce je vyvinout interaktivní aplikaci, která dokáže měřit a zlepšovat lidskou paměť, pozornost a reflexy, aniž by byla ohrožena přesnost výsledků. To vyžaduje integraci komplexních kognitivních úloh do herního prostředí, aby byl proces hodnocení zábavný a kontinuální. Kromě toho je cílem projektu poskytnout uživatelům zpětnou vazbu prostřednictvím statistik v reálném čase a sledování pokroku s využitím pokročilých technik vizualizace dat. Před zahájením realizace hry byly prostudovány nezbytné teoretické informace k pochopení procesů paměti, pozornosti a tvorby reflexů a také existující řešení této problematiky. S ohledem na jejich silné stránky a také na základě získaných informací byly vypracovány koncepty budoucí hry, z nichž byl vybrán ten nejvhodnější pro zadané úkoly.

Aplikace byla vyvinuta pomocí frameworku Flutter díky jeho široké škále funkcí pro vývoj napříč platformami a efektivnímu vykreslovacímu jádru. Pro backend byl použit systém Firebase společnosti Google, včetně správy databáze pomocí služby Cloud Firestore a služeb ověřování uživatelů. Architektura hry byla navržena tak, aby poskytovala intuitivní a poutavé uživatelské prostředí, které podporuje časté používání a udržení uživatelů.

Hra obsahuje různé kognitivní úkoly zaměřené na testování paměti, pozornosti a reflexů. Tyto úkoly jsou integrovány do hry, což zajišťuje, že je uživatel neustále vystavován výzvám. Dynamická správa stavů a životní cyklus widgetů Flutter se používají k zajištění hladkého chodu aplikace na různých platformách, což poskytuje bezproblémový uživatelský zážitek. Ke zpracování a ukládání dat v reálném čase se používá úložiště Firebase Firestore, které umožňuje okamžitou zpětnou vazbu a sledování osobního pokroku.

Aplikace obsahuje několik herních modulů spojených dohromady. Paměťový test spočívá v tom, že si hráč zapamatuje správnou sekvenci. Pozornost se měří pomocí rozptýlení, které hráče nutí více se soustředit na správná pole hry. Reflex se měří výpočtem času, který hráč potřebuje k reakci na podnět. Adaptivní algoritmy obtížnosti upravují obtížnost úloh na základě výkonu uživatele tak, aby neustále poskytovaly požadovanou úroveň obtížnosti.

Interakce uživatele a výsledky hry jsou uloženy v cloudovém úložišti Firestore, což umožňuje efektivní vyhledávání a analýzu dat v reálném čase. Uživatelské rozhraní je naproti tomu navrženo v minimalistickém, ale informativním stylu a poskytuje uživatelům zpětnou vazbu v reálném čase, historické trendy a srovnávací analýzy, které podporují sebezdokonalování. Vyvinutá aplikace úspěšně integruje komplexní kognitivní hodnocení do interaktivního a poutavého herního prostředí. Předběžné testování ukázalo, že uživatelé mohou efektivně měřit a sledovat své kognitivní schopnosti v oblastech paměti, pozornosti a reflexů. Zpětná vazba z prvních uživatelských testů zdůrazňuje potenciál aplikace jako informativního a poutavého nástroje pro kognitivní učení.

Hráči uváděli vysokou míru zapojení díky herní povaze kognitivního testování. Zpětná vazba v reálném čase, sledování pokroku a soutěživost s ostatními hráči významně přispěly k motivaci uživatelů. Přizpůsobivost obtížnosti hry zajistila, že aplikaci mohli používat uživatelé s jakoukoli úrovní kognitivních dovedností. Vyvinutá hra plně splňuje cíle této práce, a je tedy úspěšná.

Game Development for Assessment of Human Memory, Attention and Reflexes

Declaration

I hereby declare that this Bachelor's thesis was prepared as an original work by the author under the supervision of Mr. Aamir Saeed Malik. I have listed all the literary sources, publications and other sources, which were used during the preparation of this thesis.

. . . .
Denys Petrovskyi
May 8, 2024

Acknowledgements

I would like to express my gratitude to doc. Aamir Saeed Malik Ph.D. for his help and guidance throughout my thesis. I would also like to thank my family and friends for their emotional support.

Contents

List of Figures

List of abbreviations

ADHD	Attention Deficit Hyperactivity Disorder. 14–16
API	Application Programming Interface. 22, 27, 28
EEG	Electroencephalography. 9, 10, 23
ERP	Event-Related Potential. 3, 5, 9, 10, 23
JSON	JavaScript Object Notation. 26, 27
LTM	Long-Term Memory. 7
NoSQL	Not only Structured Query Language. 22, 23, 26, 38
STM	Short-Term Memory. 7
TOVA	Test of variables of attention. 14–16
UI	User Interface. 19, 20, 22, 33
UML	Unified Modelling Language. 26

Chapter 1

Introduction

1.1 Motivation

With the growth of technology, the amount of information that a person must process every second is steadily increasing. This also amplifies the need for good memory, attention, and quick reactions even in everyday situations. For example, a car driver not only needs to remember road signs perfectly but also keep an eye on them, the traffic, and pedestrians, and must react timely to the red lights of traffic signals, cats darting onto the road, and other unexpected events. Scientists have been studying how the brain processes information, and how human memory, attention, and reaction work for decades. Clinicians then apply the information obtained from these studies in practice. This work aims to explore and enhance human memory, attention, and reaction capabilities by developing and implementing a method to assess and improve these functions effectively and efficiently.

1.2 Topics of the thesis

This thesis will cover the following topics: Chapter 2 will provide the necessary theoretical basis for understanding the processes and mechanisms of memory, attention, and reaction. It will detail the neuroanatomy of various parts of the human brain and their roles in these processes. The types of memory, attention, and reaction, their stages of processing by the brain, as well as methods for measuring these functions, will be outlined. Additionally, it will explain what an ERP (Event-Related Potential) is, its role, and its use in measuring reaction and attention. Chapter 3 will cover existing solutions for measuring memory, attention, and reaction, discussing their advantages and disadvantages, and identifying opportunities for their improvement. Chapter 4 will describe the designs I developed for implementing this task, discussing their strengths and weaknesses. It will also detail the technical specifications chosen for the implementation, along with rationales for each selection. Chapter 5 will directly cover the solution itself, detailing what was done and how it was accomplished using specific examples. Chapter 6 will show what tests were performed along with their results, as well as the challenges that were encountered during the development of the application. Chapter 7 will briefly summarize the entire process, outlining the results achieved, what new insights I gained, and what improvements could be made in the future.

Chapter 2

Theoretical introduction

2.1 Memory

Memory is a fundamental cognitive process that involves storing information and retrieving it over time. It plays a vital role in enabling the ability to learn, make decisions, solve problems, and more.

2.1.1 Neuroanatomy of memory

To encode a memory, adjustments must occur in the connections between multiple neurons. This intricate process is known as synaptic plasticity, which governs the neurochemical interactions at synapses and encompasses changes in synaptic strength. While it is widely acknowledged that memories are distributed across the entire brain, there are some parts of it that are specifically important in handling memories, notably the amygdala, the hippocampus and the cerebellum [13].

- **Hippocampus** plays crucial role in formation of declarative memory. As multiple studies [5] conducted throughout the 20th century have demonstrated, removal of the hippocampus leads to the inability to create new memories and learn. However, in accordance with the standard model of systems consolidation, even without hippocampus brain is still able to recall earlier formed memories [23]. Additionally, the hippocampus holds responsibility for two essential memory functions: spatial memory, enabling us to recall object positions and navigate effectively, and associative memory, facilitating the connection of diverse memories with each other.

- **Amygdala** is a small structure located deep within the temporal lobe. It is responsible for emotional or fear-related memories, which tend to be more lasting. Amygdala is capable of associating stimulus with fearful event, storing it and then retrieving it in order to enhance ability to react to potential threat [15].

- **Cerebellum** although is primarily known for motor learning, it is also strongly implicated in procedural memory. Procedural memory encompasses skills that are executed automatically, such as walking, riding a bicycle, or playing a musical instrument.

2.1.2 Types of memory

Memory can be classified into two main types: long-term memory and short-term memory. These are two stages of memory storage progress, each having specific functions.

Short-term memory (STM) is a transient form of memory that usually lasts up to one minute. Its role is to temporally retain information, allowing to work with it. Additionally, STM has limited data-holding capacity. Back in the 19th century Sir William Hamilton estimated that, on average, capacity of STM is limited to about seven items. According to George Miller's „magical" number, the number of data can vary from 5 to 9 items simultaneously [12]. Information to the STM is mostly encoded acoustically or semantically and can be transferred to long-term memory.

Long-term memory (LTM) is a form of memory that stores information for longer periods of time, ranging from minutes to an entire lifetime. In contrast to STM, LTM does not have any capacity constraints. Long-term memory can be divided into several subtypes, including semantic memory, episodic memory, declarative memory and implicit memory.

2.1.3 Methods to assess memory

Short-term memory can be measured through various methods that usually present a participant some information and then evaluating the ability to recall it. This practices may or may not include cognitive tasks and are designed to measure the capacity of short-term memory. Methods to assess long-term memory can be similar to ones that are created to measure short-term memory. The difference is the time span before a participant would be asked to recall information they were given.

2.2 Reflexes

Before beginning this section, it is important to clarify what reflexes are within the scope of this thesis. The general concept of a reflex states that a reflex is a reaction of a living organism to some stimulus under the control of the nervous system. Reflex measurement is the calculation of the time that passes between the appearance of a stimulus and the corresponding reaction of the organism to it. Since reaction time refers to the interval of time between the presentation of a stimulus and the initiation of a response by an individual, for the purposes of this thesis these concepts are the same and both will be used later.

2.2.1 Neuroanatomy of reaction time

This process is multifaceted and can be divided into four distinct stages [18]:

1. `Receiving a stimulus`: This is an initial stage of reaction which begins with reception of a stimulus. This step is heavily reliant on the sensory systems, such as vision, hearing, and touch. For example, in case of visual stimuli, after signal from eye retina is being transited to the brain, occipital lobe receives it. The occipital lobe plays a crucial role in processing visual information, including the detection of shapes, colors, and motion.

2. `Processing the stimulus`: Once the sensory information reaches the brain, it undergoes intricate processing in various regions. For visual stimuli, the processed data is relayed to other areas like the parietal and temporal lobes, which help with spatial perception, object recognition, and memory integration. In the case of auditory stimuli, the auditory cortex, located in the temporal lobe, processes sound information, distinguishing between different pitches, frequencies, and spatial locations. This and the previous stage take around $100 - 200$ milliseconds.

3. `Making decision`: Crucial role in this stage plays prefrontal cortex. It is responsible for reasoning, judgment, and decision-making. Additionally, the amygdala can also influence decision-making, as it processes emotional responses to stimuli, potentially impacting the choice of response. The whole process of decision making requires 150 – 500 milliseconds in average, depending on complexity of the stimuli.

4. `Response`: Once a decision is made, the motor parts of the brain, such as the frontal lobe's primary motor cortex, transmit signals to the appropriate muscles and motor neurons. These impulses trigger the bodily response, which could be a quick movement, vocalization, or any other activity relevant to the stimulus. This stage executes in about 150 milliseconds and can significantly affect the entire reaction time.

The measurement of reflexes can also be significantly influenced by various factors, including fatigue, attention, and alcohol consumption. It is imperative to take these factors into account for an accurate assessment of reaction time.

2.2.2 Methods to measure reaction time

Matthäus Hipp pioneered one of the initial techniques for gauging response time in 1852, introducing the Hipp Chronoscope. As could be seen in Figure 2.1 Hipp Chronoscope employs a mechanism involving mirrors and rotating drums. Although the Chronoscope was designed earlier by Charles Wheatstone, Hipp was the one to improve first model several years later [16]. The process of measuring participant's reaction time begins with presenting a stimuli and the drum with mirrors inside of it begins to move. As it rotates, the mirror reflects light onto a screen, creating a spot of light that moves along the scale. The participant stops the measurement after perceiving the stimuli by a button or any other mechanism. The result is being evaluated according to the distance spot of light traveled.

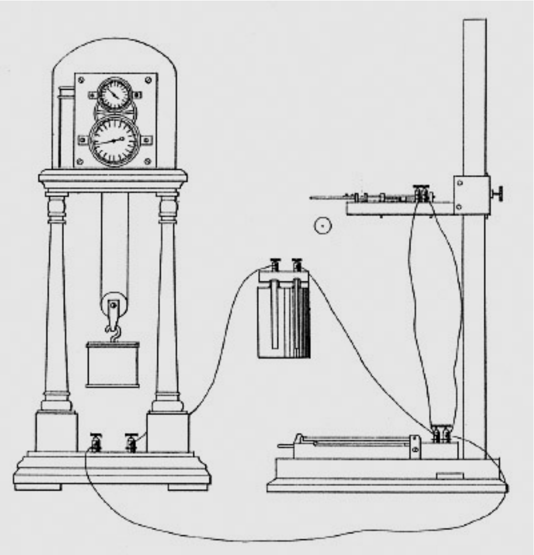


Figure 2.1: Hipp chronoscope [16].

There are numerous approaches to assess reaction time, each sharing a common concept but differing in their execution. Individuals are instructed to execute a simple action upon receiving specific stimuli. The difficulty of the test can fluctuate based on the variety of stimuli, their nature, complexity, and the intricacy of the required action. While more intricate methods for measuring reflexes may yield superior results in specific scenarios, it is noteworthy to highlight that, for obtaining accurate results, it is advantageous to minimize cognitive tasks and simplify the testing procedure.

Regardless of the method chosen to measure reflexes, a person's brain signals can be read while taking the test to get more accurate information. These signals are then processed and further analysed using event-related potentials (ERPs). ERPs are small electrical voltages generated in a specific part of the brain as a response to some stimuli. They are being measured by electroencephalography (EEG) and frequently used in different fields of neuroscience and psychology to research various processes, including reaction time.

ERPs consist of different types of waveforms, where each of them takes a specific time of reaction to the stimuli, see 2.2. These waveforms are named by a character and a number which specifies an approximate time after the stimuli when waveform reaches its peak. In Figure 2.2 character "P" stands for a positive waveform, "N" – for negative and "C" – for waveforms that do not have a specifically assigned polarity [10].

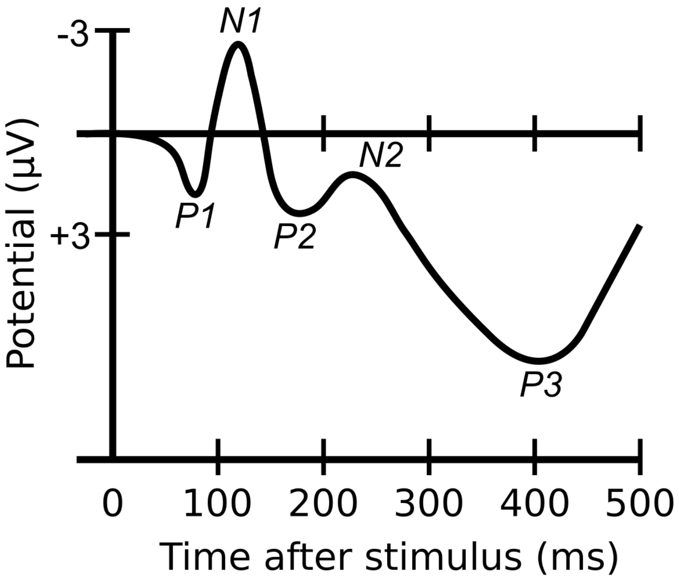


Figure 2.2: ERP waveforms [21].

In terms of reflexes, due to approximate time costs of reaction time stages, main waveforms to be focused on are: C1, N1, P2, N2, P3. Different ERP waveforms allow to better understand individual's test data. In case a deviation from the norm is detected, ERP analysis provides a specific timeline where processes were executed slower than usual. This information can help identify possible illnesses, prompting people to seek appropriate medical guidance.

2.3 Attention

Attention is an ability to process information while tuning out other details. Attention allows individuals to focus on a single stimulus, which is essential for processing incoming information effectively given the brain's limited capacity. Attention can be split into several types [8]:

- Sustained Attention: This type of attention requires maintaining focus on a single activity for a prolonged period without getting sidetracked. It is often needed for tasks that demand continuous involvement until they are fully completed. Additionally, it is noted that sustained attention typically diminishes as individuals age.
- Alternating Attention: This involves the ability to switch focus between multiple tasks that have varying cognitive requirements, rather than trying to attend to several tasks at once. It necessitates cognitive flexibility to effectively switch and manage different activities within short intervals.
- Selective Attention: This form of attention is about the capacity to selectively concentrate on specific elements while ignoring others. It entails giving priority to certain aspects within the environment and excluding non-pertinent stimuli. This requires the ability to ignore both external interruptions and internal diversions to remain focused on a specific task.
- Focused Attention: This type of attention is triggered by sudden stimuli, such as a sudden loud sound. Responses to these stimuli are immediate and swift. Focused attention is essential in scenarios where a quick reaction is necessary.
- Limited Attention: Relates to the ability to multitask by dividing attention across several tasks simultaneously. This differs from alternating attention by the requirement to focus on multiple stimuli at the same time.

2.3.1 Neuroanatomy of attention

Since attention is very complex cognitive task, it requires many different brain structures, understanding which is extremely important for future parts of the research. Although usage of brain can differ based on the type of attention that is being currently used, there are key parts involved in neuroanatomy of attention. Prefrontal cortex is extremely important in controlling attention as it is involved with processes of goal setting and decision-making. Partial lobe is essential for selective attention as well as allocating new attention resources when a new stimuli is being received. Visual cortex of an occipital lobe is responsible for processing visual information and can contribute to selective attention of visual stimuli. It works together with temporal lobe in detection of new stimuli and reorientation of the attention.

2.3.2 Methods to test attention

In order to measure individual's attention different cognitive tasks exist. These tests cover various types of attention and help in detecting different diseases connected with attention. To receive data about individual's attention EEG can be used, as it is a great tool to study electrical changes during attention. Also, event-related potentials (ERPs) are useful to further process the data. With ERPs focus should be shifted to the first negative peak that occurs in 100ms after stimuli appeared. According to the time of occurrence of N1 peak (see Figure 2.2), can be found out how advanced individual's attention is.

Chapter 3

Existing solutions

In this chapter already existing games that measure and assess memory, reflexes and attention will be described.

3.1 Games to assess memory

An illustrative instance of a game assessing an individual's short-term memory is the one formulated by CogniFit [1]. As expounded upon in an earlier chapter this method is based on the principle in which the participant is presented with various information, in this case, pictures, see. While sequence of images is being presented, individual has to press space bar every time he notices am image that repeats. This approach is effective not only in assessing the capacity of short-term memory but also in enhancing its capabilities.

Washington.edu offers another instance of a straightforward yet impactful technique [19]. In this method, participants are presented with sequences of letters divided into multiple segments, with the objective of recalling as many letters in each segment as feasible. While the typical range for short-term memory capacity fluctuates between 7 and 9 pieces of information, washington.edu presents segments comprising 12 letters concurrently. This phenomenon is attributed to the human tendency to perceive vowels and consonants as a unified element, impacting memory retention. The simplicity of this exercise enables a heightened focus on memory, minimizing cognitive task interference. The biggest drawback of this testing method lies in the insufficient provision of additional test cases for letter sequences. According to this information this game is suitable only for one-time testing.

In September 1913, Knox introduced an alternative approach for evaluating short-term memory. In this method, the examiner used a pencil to make contact with four objects, such as cubes, and the participant was required to reproduce the sequence in the exact order of presentation, see Figure 3.1. Many researches have shown that not only this test can show how good individual's memory is, but also their overall intelligence. Later this test was numerously modified by increasing or decreasing amount of cubes in the sequence gradually in order to make this method more precise and adapt difficulty according to participants performance [14]. In Table 3.1 will be described advantages and disadvantages of previously mentioned solutions.

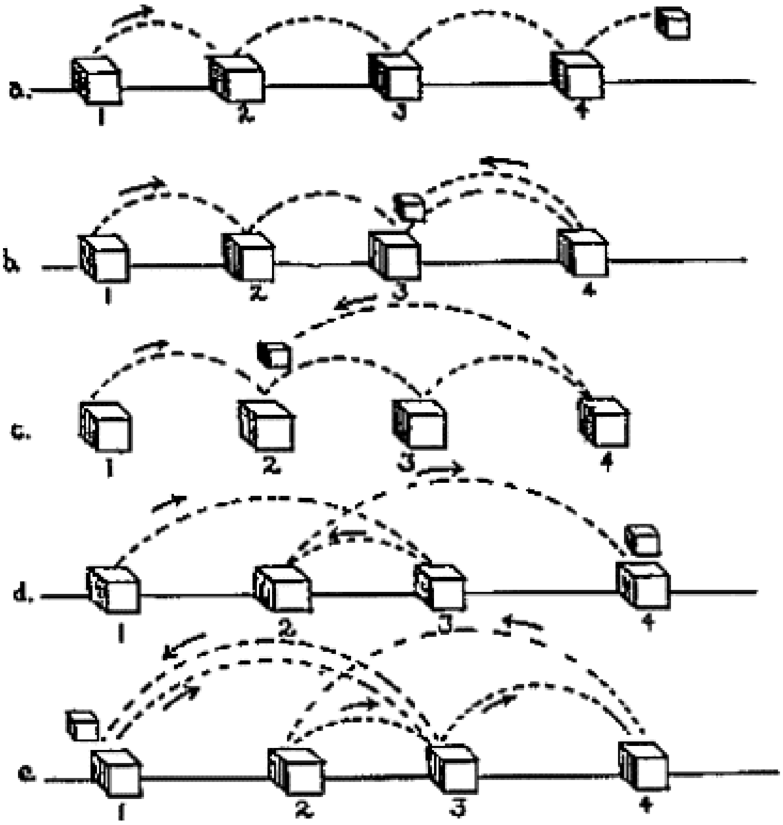


Figure 3.1: Knox's cubes [14].

Table 3.1: Pros and Cons of existing memory testing games.

	Advantages	**Disadvantages**
CogniFit's Image Recognition Game	• Effectively assesses short-term memory capacity. • Enhances short-term memory capabilities. • Uses visual stimuli, engaging participants in a dynamic way. • Involves active participation with the space bar, making it interactive.	• Limited information provided on the specific mechanics and scoring system. • May be more suitable for repeated testing rather than continuous assessment.
Washington.edu's Letter Sequences Game	• Simple and straightforward design for easy administration. • Focuses specifically on memory without significant cognitive interference. • Challenges participants with larger-than-average letter sequences.	• Insufficient provision of additional test cases for letter sequences. • May be more suitable for one-time testing rather than repeated assessments.
Knox's Object Recalling Game	• Evaluates short-term memory and overall intelligence. • Can be adapted for precision by modifying the number of cubes. • Offers a physical and tactile component, engaging different senses.	• Limited information on specific modifications and adaptations made. • Requires physical objects, making it less convenient for remote testing. • The tactile nature might introduce variability based on individual sensory preferences.

3.2 Games to assess reflexes

A straightforward and efficient approach to measure an individual's reaction time is demonstrated by the „Human Benchmark“ [6]. This method gauges reaction time by assessing the duration it takes for the participant to press a button in response to a color change on the screen. Because of the test's simplicity, it offers the opportunity to eliminate any variables that could impact the outcome, enabling a focused assessment solely on reaction time.

A more creative way to assess response time can be seen in the game created by JustPark. The task is simple – to press a button when a stimuli occurs on the screen. What makes it different is the user interface of the game. It is implemented in the way where the individual, who is passing the test, is driving a car and they have to stop it the time they see a corresponding road sign. Although this game design may drag away individual's attention from the stimuli, it encourages the player to repeat test more [7]. The following Table 3.2 will describe pros and cons of previously discussed games.

Table 3.2: Pros and Cons of existing reflexes testing games.

	Advantages	**Disadvantages**
Human Benchmark's Color Change Reaction Time Test	• Simple and straightforward design for easy administration. • Eliminates variables that could impact the outcome, ensuring focused assessment. • Provides a clear measure of reaction time based on a color change. • Suitable for quick and repeated testing.	• May lack engagement compared to more interactive or visually stimulating tasks. • The simplicity of the task may not reveal nuanced aspects of reaction time.
JustPark's Car Reaction Time Game	• Creative and engaging game design with a realistic scenario. • Encourages repeated testing by making the task enjoyable. • Provides a contextually rich environment for reaction time assessment. • Mimics real-world scenarios.	• The game design may distract participants from the primary stimuli. • Variability in attention due to the engaging interface may affect results. • The immersive nature of the game may introduce additional cognitive factors.

3.3 Games to assess attention

The Stroop color and word test is a well-recognized exercise that emphasizes selective attention and demands considerable cognitive adaptability. This effect was first described in 1930s by psychologist John Ridley Stroop [11]. In this task, color words are displayed in ink of a different color, and participants are instructed to identify and vocalize the ink color while disregarding the written word.

The Flanker test shares similarities with the Stroop test in its emphasis on selective attention. Participants are tasked with concentrating on a target stimulus while disregarding non-target stimuli.

The game that assesses sustained attention is called “The test of variables of attention„ (TOVA) and it was introduced in 1960s by Dr. Lawrence Greenberg. This test usually lasts 21.6 minutes for adults and is usually used to identify people with ADHD. This method requires participant to respond as quickly as possible to a stimuli that appears at regular intervals. Sometimes a non-target stimuli may be included in the test [22]. In Table 3.3 will be shown their advantages and disadvantages.

Table 3.3: Pros and Cons of existing attention testing games.

	Advantages	**Disadvantages**
Stroop Color and Word Test	• Well-recognized and widely used for assessing selective attention. • Emphasizes cognitive adaptability and inhibitory control. • Provides a clear measure of the interference between color and word processing. • Sensitive to cognitive impairments and attentional deficits.	• May be susceptible to individual differences in language proficiency. • The task's complexity may make it challenging for certain populations. • Potential for practice effects with repeated administrations. • May not fully capture sustained attention, focusing more on interference control.
Flanker Test	• Similar to Stroop test, emphasizes selective attention. • Helps assess the ability to concentrate on a target while ignoring distractions. • Can provide insights into cognitive flexibility and inhibitory control.	• May not fully capture the complexity of real-world attention demands. • Limited variability in task design compared to more dynamic attention assessments. • Potential for participants to develop strategies with repeated testing. • Less emphasis on cognitive adaptability compared to the Stroop test.
Test of Variables of Attention (TOVA)	• Specifically designed for assessing sustained attention. • Utilized for identifying people with ADHD. • Involves responding to stimuli at regular intervals, mimicking real-world attention demands. • Objective and standardized, reducing potential for bias.	• Long duration (21.6 minutes) may introduce fatigue effects. • Primarily designed for identifying ADHD, limiting generalizability to other attention disorders. • Limited in providing detailed insights into specific attentional components. • May not capture the full range of attentional processes.

3.4 Existing solutions' summary

Various existing games designed to measure and assess memory, especially short-term, reaction time and attention were examined. Each of them presents unique approach, advantages and disadvantages, contributing to research of cognitive assessment methodologies.

In terms of games that assess memory, CogniFit's Image Recognition Game stands out since it can be used not only to test but also to improve short-term-memory using visual stimuli. The Letter Sequence Game provided by Washington.edu is a straightforward yet effective way to measure individual's memory. Knox's Object Recalling Game provides assessment of general intelligence also through out tactical stimuli. Nonetheless, disadvantages include limited amount of tests that a single individual can take and the need for real items.

In context of games that measure reaction time, the Color Change Reaction Time Test from Human Benchmark offers a simple method that removes variables to guarantee a focused evaluation. The immersive design of JustPark's Car Reaction Time Game may bring possible distractions despite its creative engagement. Despite each method having its own disadvantages, they prove themselves to provide accurate results though simple tests.

As for games used to evaluate attention, the Stroop Color and Word Test and the Flanker Test are useful tools for assessing inhibitory control, cognitive flexibility, and selective attention. Sustained attention is the focus of the Test of Variables of Attention (TOVA), which is especially useful in diagnosing ADHD. But every test has its own set of difficulties, such as problems with language competency, the difficulty of the task, and the inability to provide comprehensive insights into particular attentional components.

Chapter 4

Proposed solution

During my research, I came up with several ideas for implementing a game that would test memory, reaction and attention, but ultimately settled on the following game design.

4.1 Game concepts

First idea is to create a game that will consist of a field with equal squares 4x4 (see Figure 4.1). Three random squares will light up with different colors (red, green and blue) every short period of time. This action, according to the capacity of human's short-term memory, will repeat up to 9 times, as shown in Figure 4.2. After that, player will be asked to recall the sequence of blue squares, the red and green ones are there for a distraction affect, that helps to assess alternating and selective attention, which was detailed in 2.3. The results of tests will be stored not only to present them to a player, but also to adapt the difficulty of the test itself. In order to be successful in this game individual will require not only a good short-term memory, but also an exclusive reaction time and attention. Also the color based feature of the game makes it more exciting, which encourages player to repeat the test without getting bored.

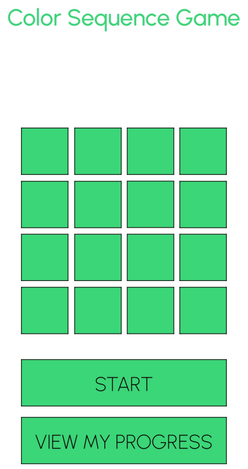


Figure 4.1: First design of concept #1.

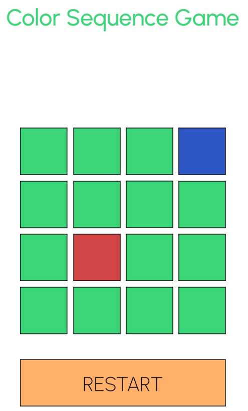


Figure 4.2: First design of concept #1.

The other suggestion is the game that involves presenting player a sequence of symbols, numbers, or images displayed on the screen one by one, with varying time delays that adjust based on the user's progress, see Figure 4.3. Following the sequence, the player will be prompted with questions related to the sequence, such as determining the number of items or calculating the sum of numbers, see Figure 4.4. The difficulty of the game, including the length of the sequence and time delays, will be automatically adjusted based on the player's performance. In addition, the game aims to improve attention by introducing distracting elements within the sequence that the player must ignore. As in the previous concept data will be stored for future analysis and making adjustments to the difficulty. Although this game design introduces cognitive tasks that may impact the final result, the game provides wide range of content options, including words, numbers, images and questions, that will raise the engagement.

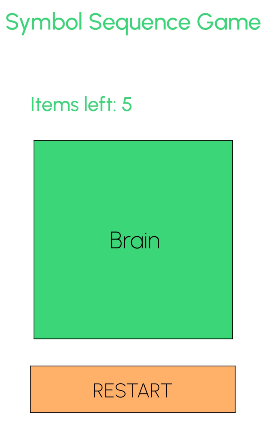
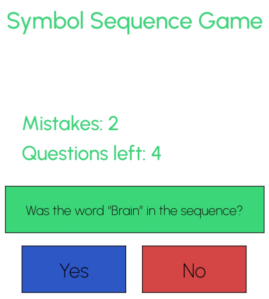


Figure 4.3: First design of concept #2.



Was the word "Brain" in the sequence?

Figure 4.4: First design of concept #2.

The final concept, despite having similar approach with the Stroop test, assesses besides attention also short-term memory and reaction time. The game entails displaying a sequence of words, where each word represents a specific color but is visually presented in a different color (for instance, the word "blue„ appearing in yellow), see 4.5. These words will be introduced with varying time intervals, adapting in response to the user's progress. Following the sequence, the player is required to associate each word with the color in which it was displayed 4.6. The player's advancements will be logged, and the length of the word series will be dynamically adjusted based on the individual's performance data. This suggestion provides its own engaging way to measure memory, reaction time and attention also integrating cognitive tasks. The main downside is that players may find this approach to challenging due to requirements of color recognition and word matching skills simultaneously.

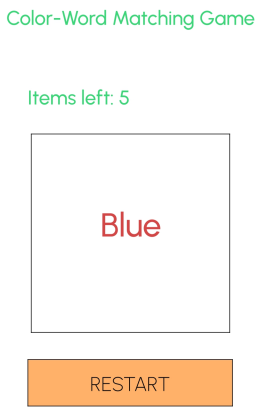


Figure 4.5: First design of concept #3.

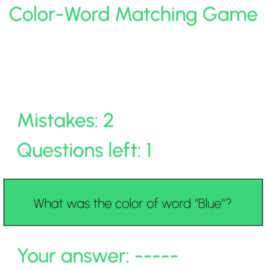


Figure 4.6: First design of concept #3.

According to all the advantages and disadvantage of aforesaid ideas as a design of the game was chosen the first concept by the reason of it providing not only great challenge to memory or attention but a reaction time also. Color Sequence Game does not include difficult cognitive task as other two suggestions do. Additionally final project will include possibility to create and log into account for storing data and progress analyses.

4.2 Technical specifications

This section will discuss software stack applications, their advantages and the reasons why they were chosen.

4.2.1 Flutter

To implement the application with the aforesaid design was chosen framework **Flutter** for several reasons. Flutter is an open-source UI software development kit created by Google. Applications in it are written in the `Dart` language, which allows for convenient testing using just-in-time execution during development without sacrificing performance.

Widgets

In Flutter, the fundamental building block of the user interface is the widget. It is the foundation in which everything from simple lettering to complex designs and everything in between is implemented. Understanding the concept of the widget and the widget tree is essential to building an application in Flutter. All widgets are objects or instances of the 'Widget' class and also have no visual representation, just show how different parts of the interface should be arranged on the screen based on their own configurations and state. Widgets have two types:

- `Stateless Widget` is a widget that does not depend on any date state to render itself. Although their state can change based on external factors, they themselves do not process the state internally. An example is the 'Text' widget, which is a simple string of text.

- A Stateful Widget is a widget that maintains its state, which can change throughout its lifetime. For this purpose it creates an instance of the 'State' object that rebuilds the widget in case of a state change using State.setState. Such widgets are useful when the user interface must change depending on any external or internal factors affecting its state [3]. A simple example is 'Checkbox', which changes in case of user interaction with it.

Widget tree

Also in Flutter widgets have their own hierarchy called widget tree. It shows the structure of widgets to build the desired UI, where a widget is nested inside its parent widget. The top-level widget is often the application itself, as represented by widgets such as 'MaterialApp', their job is to give the framework the basic styles of the application. Inside them there will be other widgets that display e.g. navigation, page content, etc. Widget life cycle consists of 4 parts:

1. The framework calls the widget constructs thus creating it.
2. Adding the widget to the element tree.
3. Updating the widget, in case of changes in its configurations or dependencies.
4. Destroying the widget and then removing it from the element tree.

Rendering in Flutter

When compiling a Flutter application, the framework creates an element tree by traversing the widget tree (see 4.7). Its main task is rendering, it holds the widgets and updates them to the current state. The elements in this tree are instances of widget configurations. Since widgets themselves are immutable and do not hold information about parent and child widgets, this tree is necessary to maintain the structure of the original interface. Nodes of element tree can be of two types: ComponentElement and RenderObjectElement. The last one represents an element of UI, while the first one serves as a host for other elements.

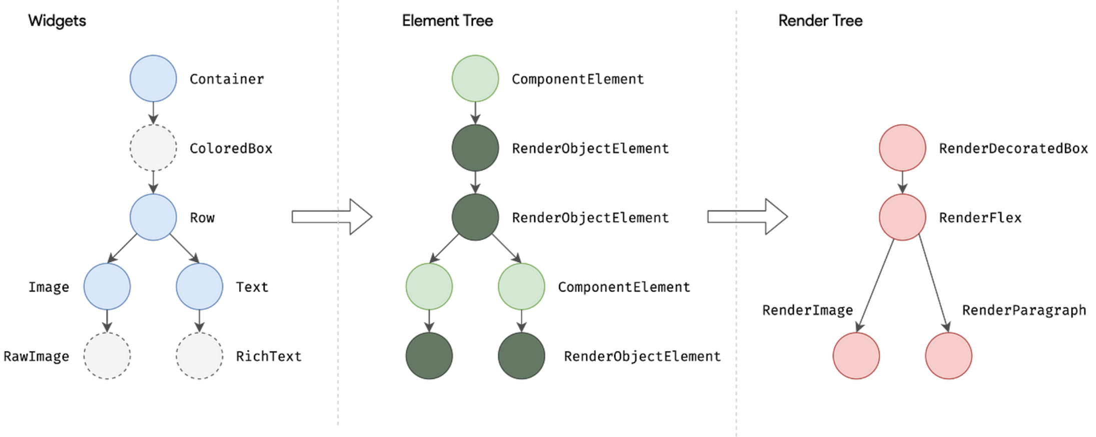


Figure 4.7: Example of flutter trees [2].

The basis of rendering and its pipeline is the `render tree`, which is a collection of `RenderObjects` that is created for each RenderObjectElement from the element tree (see 4.7). Each RenderObject knowing about its parents and children defines a model that helps with the location and rendering of the object on the user's screen [2]. When Flutter needs to render an object, the `'build()'` method is called, which returns a subtree that need to be rendered based on the current state of the application. Rendering is done walking through a render tree with a `'depth-first' approach` (see 4.8, which means that the framework will go through each branch of the tree starting from the root to the deepest cells, thus before starting the next branch, the previous one is fully processed first.

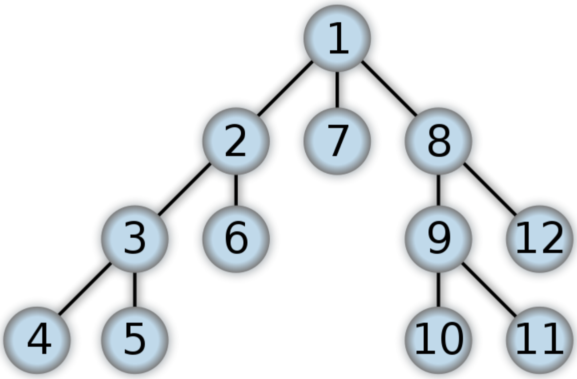


Figure 4.8: Depth first search tree [20].

Flutter's rendering engine is made for efficient rendering of the user interface and is based on the `Skia` graphics library. Skia is an open-source library with high performance, which is used not only by Flutter but also by Google in general. It is responsible for all low-level operations, such as rendering different parts of the interface, which are passed by Flutter using commands. The rendering process in Flutter consists of several stages and is called rendering pipeline.

- `Layout Phase`: at this stage `RenderObjects` calculate their size based on parental constraints, so that not only the size of the objects but also their position on the screen is known. This process is recursive throughout the entire rendering tree.
- `Paint Phase`: after the end of the previous phase, the rendering phase begins. Using the Skia engine, the objects are drawn on the canvas, after which the instructions from the canvas are converted into pixels and transferred to the screen.
- `Compositing Phase`: this phase consists of combining the different layers into a finished image on the screen. During this phase, various image effects such as transparency are also processed.

Advantages

Based on the above-mentioned characteristics and features of the Flutter framework,main advantages of Flutter and the reasons why it was chosen as the framework for my application can be pointed out.

- The first advantage of Flutter is its power and efficiency in rendering thanks to the use of the `Skia` engine, which is famous for its high high-performance capabilities. This engine allows rendering complex UIs while maintaining a high frame rate, which is crucial for an application aimed at measuring memory, attention and reaction. This is also aided by Flutter's rendering pipeline. By determining the position of objects on the screen in the first phase and rendering them in the second phase, the UI is rendered smoothly and optimised.

- Flutter allows you to build an extensive and clear interface without complicating the development process thanks to a wide range of widgets. And the division of widgets into stateless and stateful allows to effectively monitor the state of the user interface and update it if necessary, thus improving the responsiveness of the interface and making it user-friendly.

- One of the main advantages of Flutter is its ability to be cross-platform with a single codebase. This feature removes the need to write code specifically for each platform, thus reducing the time and resources needed for development. Additionally, Flutter simplifies development with various features such as hot reload, which allows you to see changes in the application without having to restart it. This feature directly speeds up the processes of fixing bugs or user interface.

4.2.2 Firestore

In application will be used **Firebase** for it's compatibility with Flutter and it's extensive range of functionality. Among these `Cloud Firestore`, `Firebase Authentication` are especially important for the application infrastructure, as detailed further in 5.3 and 5.5.4. To bridge Flutter application with Firebase set of plugins called **FlutterFire** will be utilized. FlutterFire not only simplifies integration of Firebase but also enhances development providing unified API that utilizes powerful features of Dart language. FlutterFire allows seamless development ensuring that both frontend and backend parts of the application are fully integrated.

Cloud Firestore

Cloud Firestore, is a convenient database with many benefits. As Firestore is a NoSQL database, data is stored in it in the form of collections and documents, where collections are storages for documents, and documents consist of fields that store different data.

NoSQL database implies an extensive category of database management that is different from the traditional database. These databases are designed to be able to store a large number of date models. They are known for their ability to flexibly manage structured and unstructured data and perform best when large amount of dates need to be transferred with minimal latency. Among the main advantages of NoSQL databases are:

- `Scalability`: unlike SQL databases, which can be scaled vertically by adding by expanding the server capacity by adding memory, NoSQL databases can be scaled horizontally, which means that such databases can be scaled to multiple servers simultaneously [17].

- `Flexibility`: NoSQL databases are flexible and their schema can be changed without much effort during development, which is a great simplification of application development where new fields with information can be added to the database any time.

- `Performance`: Since the NoSQL database schema is generally simpler, it allows you to get answers to queries and access to data faster. This ability makes them most suitable in cases of need for fast real-time data updates.

In case of Cloud Firestore it has several advantages why it was chosen. Its first advantage is its document-orientation, storing data in documents and collections with a large set of datatypes allows data to be easily transferred directly to objects in code. One of the main advantages of this database is its ability to synchronise data between the user and the cloud in real time. In addition, Firestore allows you to store data in cache when there is no internet connection, as soon as the user connects to the network, the data from the cache will be updated.

4.3 Assessment from direct signals

To obtain more accurate data about the user's attention and reflexes, it is possible to read their brain signals while playing the game. The EEG headset from OpenBCI can be used for this purpose for several reasons, see 4.9.

- A 16-channel configuration provides good spatial resolution and allows more comprehensive monitoring of brain activity compared to lower-channel systems.
- OpenBCI provides open-source software tools that can be used data acquisition, signal processing, analyses.
- Being part of open-source community, OpenBCI headset receives strong community support that can help with troubleshooting and finding solutions.

Once the data is obtained, ERPs can be extracted from it for future analysis. In the case of attention and response studies, the main peaks to focus on would be P50 for attention and C1, N1, P2, N2, P3 for reflex analysis. For more details, see 2.2.2.

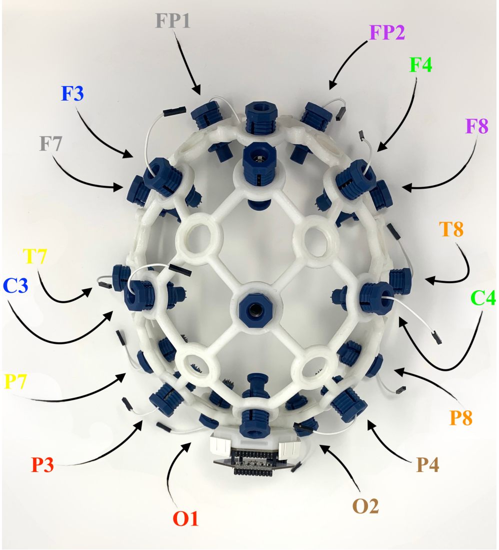


Figure 4.9: Headset from OpenBCI [9].

Chapter 5

Implementation

This chapter shows implementation of my solution proposed in the previous chapter. The application implementation is also stored on `GitHub`, where you can find the current version of the application[1].

5.1 File structure

The file structure of my solution is largely provided by Flutter framework, which prescribes specific organization of files and directories as well as provides with configuration files for "basic„ characteristics. Source code of the application is being stored in folder `lib\`. Folder `assets\` is designated for images used in the application. Although application was created specifically for Android platforms, folders `android\`, `ios\`, `macos\`, `linux\`, `web\`, `windows\` contain platform-specific code and resources to build application, for Android, IOS, macOs, Linux, web and Windows respectively. Folder `build\` is being generated by Flutter during build process and contains compiled versions of the application. `README.md` contains text documentation of the app. Additionally, high-level architecture of the application includes some files and folders that are specifically designed to configure tools used in the project. Among these are:

- **analysis_options.yaml** specifies the project's analysis settings for Dart code, helping maintain code quality and consistency.
- **firebase.json** is a part of Firebase configuration, detailed further in 5.2.
- **functions\** also is responsible for Firebase configuration, in particular for cloud functions.
- **node_modules\** contains Node.js packages used by Firebase functions.
- **package.json** and **package-lock.json** are parts of Node.js configurration specifying dependencies and their versions.
- **pubspec.lock** is an auto-generated file that ensures project uses specific versions of each package to keep builds consistent.
- **pubspec.yaml** is a main file for configuration of the project, where dependencies and assets are defined, for more details see 5.2.

[1] `https://github.com/HelHeim111/VUT-IBS`

5.1.1 Description of source files

While each source file was designed to implement a specific scope of tasks, they are highly depend on each other. This includes not only switching pages of the application but also sharing data through the database. Description of each source file:

- **main.dart** is the point from where the application starts. It holds Firebase initialization and routes to every page of the application.
- **firebase_options.dart** holds options for Firebase configuration on different platforms and was provided with configuring Flutterfire.
- **home_page.dart** is the main page that user sees right after opening the application. More in 5.5.1.
- **navigation_bar.dart** is responsible for the navigation bar, which the user will see at the bottom of the screen, to quickly and easily move the user between the pages of the application. Detailed in 5.5.3.
- **sign_in.dart** allows a user to sign in to an existing account. Detailed in 5.5.4.
- **sign_up.dart** is responsible for allowing the user to create a new account. Detailed in 5.5.4.
- **begin_game.dart** is one of the most important files of the application, as it implements the functionality of the game that was proposed in the previous chapter. For more details see 5.4.
- **user_statistics_page.dart** is responsible for the page that shows user statistics, results and progress in form of charts. More in 5.5.2.

5.2 Setup of dependencies

During the development, a set of various dependencies and configurations was selected for the implementation and maintenance of the application's functionality. The primary dependencies are described in the `'pubsec.yaml'` file, and configurations in `'firestore.json'`.

As stated earlier, 'pubsec.yaml' acts as the cornerstone for managing the project's dependencies. External libraries that are necessary to expand the application's functionality must be listed in this file. The declaration of these dependencies allows for the version of each package to be controlled and utilized throughout the project files.

The fundamental dependency is the `Flutter SDK`, which provides access to creating and interacting with the user interface. For the construction of the graphs described in 5.5.4, the package `'charts_flutter'`, which was connected in version 0.12.0, is needed.

Equally important to the Flutter SDK are the Firebase-related packages, namely: `'firebase_core: 2.27.1'`, `'firebase_auth: 4.17.6'`, `'cloud_firestore: 4.15.9'`. While the first serves to initialize Firebase in the project, the following two are responsible for Cloud Firestore and Firebase Authentication, respectively.

The package `'flutter_launcher_icons: 0.10.0'` simplifies the customization of the app's icon, which was also connected in this file in the 'assets' section. In addition to the logo, this section is responsible for all the media files used in the application. The `'connectivity_plus: 2.3.7'` package was added to check whether the user utilizing the application is connected to the network. It allows for detecting and adapting to different network states.

The 'firebase.json' file complements 'pubsec.yaml' and holds the configurations for the Firebase service. The most important part of the configurations is the 'functions' array which points to Firebase Cloud Functions. The array stores the path from where the functions will be deployed, as well as the commands that will run before the functions are deployed.

5.3 Database architecture

In my application, Cloud Firestore, which was covered in 4.2.2, is used to save data about users, their games and progress. This information is then retrieved from the cloud to customise the difficulty of the game and also to display statistics.

Since Cloud Firestore is a NoSQL database, it makes no sense to represent its structure using UML diagrams, because documents can have a different set of fields. To represent the structure of collections and documents inside the database it is better to use JSON tree, see Listing 5.1.

```
{ //collection
    users: {
        //documents
        user1_ID: {
            //collections
            games: {
                //documents
                2024-04-13 21:04:26.492320: {
                    //fields
                    at_mistakes: (number),
                    average_reaction_time: (number),
                    mistakes: (number),
                    sequence_length: (number),
                    timestamp: (timestamp),
                }
                 2024-04-13 21:35:02.443526:{...}
            }
            userProgress: {
                //documents
                progress: {
                    //fields
                    age: (number),
                    games_played: (number),
                    name: (string),
                    new_sequence_length: (number),
                }
            }
            //fields
            dummy : dummy,
        }
        user2_ID: {...}
    }
}
```

Listing 5.1: Firestore JSON tree.

Collection **'users'** is a top-level primary collection. Each it's document contains information about individual user identified by theirs unique ID. All the information about user is divided between two sub-collections. Additionally, user document possesses a placeholder field named **'dummy'**. Because in Cloud Firestore, which was covered in 4.2.2, a sub-collection can be inherited even from an empty document, the 'dummy' field serves to maintain the existence of the document. If a document is missing, it is not possible to extract any data from sub-collections that are in it using **DocumentSnapshot**. The **'games'** collection stores documents with information about each game played by the user, such as date, number of errors, reaction time. **userProgress** contains a document with information about the difficulty of the next games and some personal data of the user.

5.3.1 Communication with database

In order to start communicating with Firestore package **cloud_firestore.dart** is used. It provides various tools and APIs for working with Firestore directly from the Flutter application, the most important of which for the project are Querying and Transactions, which can be used to retrieve and filter data. The following details will describe usage of database in defferent parts of the application.

To store personal user data (e.g. 'name', 'age') after creating a new account in the **'sign_up.dart'** file, data uploading to Firestore via FirebaseFirestore.instance is implemented. Using collection and document navigation, data in the form of the Map class is loaded into the database. Using a similar approach, the data is also saved after each game played by the user, see Listing 5.2. This information will allow building statistics and tracking progress.

```
void saveUserData(User user, String name, int age) {
    FirebaseFirestore firestore = FirebaseFirestore.instance;

    Map<String, dynamic> userData = {
      'email': user.email,
      'name': name,
      'age': age,
      'games_played': 0,
    };

    firestore
        .collection('users')
        .doc(user.uid)
        .collection('userProgress')
        .doc('progress')
        .set(userData)
        .then((value) {
      print('User data saved to Firestore');
    }).catchError((error) {
      print('Failed to save user data: $error');
    });
}
```

Listing 5.2: Storing data after signing up.

As for retrieving data from the database, JavaScript APIs **QuerySnapshot** and **DocumentSnapshot** were used for this purpose, where the former is responsible for retrieving all documents within a particular collection specified using a path, and the latter is responsible for retrieving the fields of a particular document, see Listing 5.3. QuerySnapshot in the application was used only when it is necessary to get information about several users at once, as, for example, when creating a leaderboard on the main page 5.3.1. In case the ID of a particular user whose information should be retrieved from the database is known, DocumentSnapshot is used. Transactions using the above APIs in case of concurrent edit are restarted from the beginning, thus protected from unexpected errors.

```
QuerySnapshot querySnapshot = await FirebaseFirestore.instance
      .collection('users')
      .get();
  if (querySnapshot.docs.isNotEmpty) {

    for (QueryDocumentSnapshot userDoc in querySnapshot.docs) {
      DocumentSnapshot progressDoc = await FirebaseFirestore.instance
          .collection('users')
          .doc(userDoc.id)
          .collection('userProgress')
          .doc('progress')
          .get();
    }
  }
```

Listing 5.3: Fetching data for leaderboard.

5.4 Game implementation

Before moving on to discuss the details of the game's implementation, it is necessary to clarify its design. Initially, the idea was to separate the testing of memory and attention from the testing of reaction. This was justified by a desire not to overload the user with actions that would be required while playing. However, splitting the game into two smaller ones brought complications not only for the player, who would have had to play both to obtain current data in all three characteristics but also would have deviated from the original goal of creating a game capable of testing memory, attention, and reaction simultaneously. Therefore, after careful analysis and searching for solutions to this problem, it was decided to combine everything into one game in the following way. After a new trio of squares changes color, the player needs to press the playfield as quickly as possible, thereby allowing the measurement of the reaction. Repeating this action for each trio of squares will calculate the average reaction time, and it will be shown to the user. More about the game's design can be found 4.1.

It is also important to clarify how a player's progress will be calculated. The calculation of reaction time follows from the above. To calculate human memory, the number of mistakes user makes when recalling a sequence of blue squares will be counted. As for attention, the user's success in this indicator will be counted based on the number of times when during the recalling the sequence of blue squares user clicks on the field that had one of the other two colours: green or red. This will show when the user was distracted by the other colour and lost attention.

Upon opening the game page user is greeted with interactive page that includes game manual, see Figure 5.1. This manual was designed using simple yet effective a `Widget` that returns a `Container` with the instruction. In case of an error when loading the manual, it returns an empty Container. This approach assures that application stays robust and does not crash when encountering such an error.

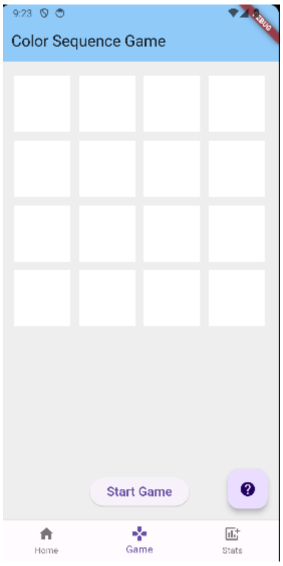


Figure 5.1: Game page before starting playing.

To start the game, the user must press the 'Start Game' button. The first thing that happens after starting the game is calling the asynchronous function `startTimer()`. Before the main functionality of the function, the user data is obtained using the function `fetchStatisticData()` with the keyword await, this prevents the possibility of starting the game without actualising the data by suspending the game until the end of the function execution. Next, the `startTimer` function starts a two-second timer, at the end of which three different squares are randomly selected from the field. Each square is assigned a different color and they are written into arrays. Using the `setState()` method, the status of the Stateful Widget is updated and color of squares is being changed, see Figure 5.2. If the maximum length of the sequence of squares (`stmThreshold`) has been reached, the field returns to its original state with a delay of one second.

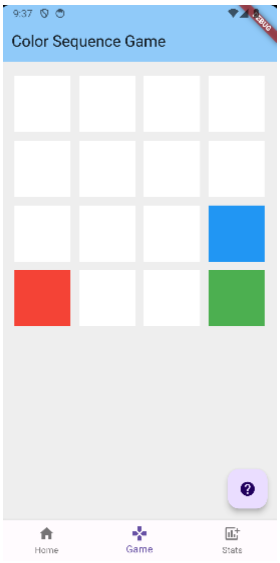


Figure 5.2: Game page during gameplay.

During the game, the user will also tap on the screen every time he notices a field update. This action is necessary to calculate his reaction time. The timer that measures the reaction starts when the new squares are assigned their colours and since measuring the reaction requires extreme accuracy, the time it took for the stimulus to appear is subtracted from this time.

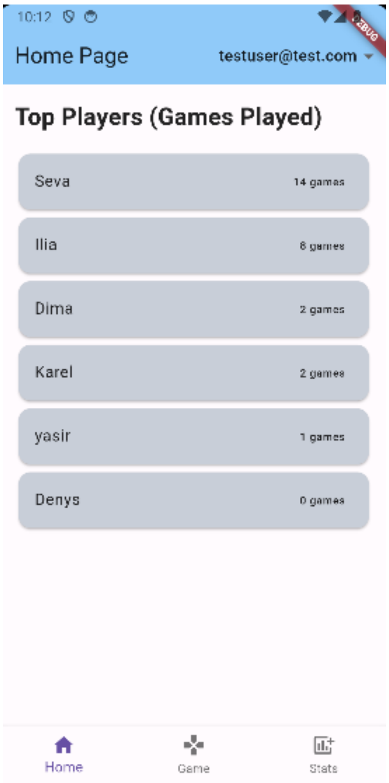
After finishing the sequence of squares, a stage starts where the user has to recreate the sequence. With each click on the playing field, the status of the widget is updated. At the end of the game, the user's errors are calculated by comparing the arrays with the correct sequence and the user's sequence. The average reaction time is also calculated. Finally, all results are shown on the screen.

5.5 Additional Components

5.5.1 Home page

At the beginning of development the home page was supposed to be just for greeting the user and moving him to other pages, but later it was decided to add a competitive part to the game for more user involvement. For this purpose on the home page was added a table of leaders among players, which shows the top users by the number of games played by them, see Figure 5.3.

For the realisation of this idea was used widget `SingleChildScrollView`, which consists of widgets `Card`, each of which has information about the user. Each time a new user visits the homepage, the user data is retrieved from the database, see Listing 5.3.1, thus updating the content of the leaderboard.



Home Page testuser@test.com

Top Players (Games Played)

Seva	14 games
Ilia	8 games
Dima	2 games
Karel	2 games
yasir	1 games
Denys	0 games

Figure 5.3: Home page.

Additionally, in the upper right corner of the home page,buttons to access page for registering a new account or logging in to an existing account if the user has not yet logged in have been added. In the opposite case, a drop-down list appears in the same place, which provides information about who is playing and allows the user to log out of the account.

5.5.2 User's statistics

As mentioned above 5.1.1, the statistics page stores information about the user's progress over time in the form of charts. Upon entering the page, the user has the option to select one of three categories: reaction time, memory, or attention, after which a graph corresponding to the selected category will appear on the screen, see Figure 5.4. The creation of these graphs requires the package `<charts_flutter/flutter.dart as charts>`. Once the latest information about the user has been retrieved from the database, its processing begins. To avoid overloading the graph with information, the results of games played at the same time are displayed as a single average value.

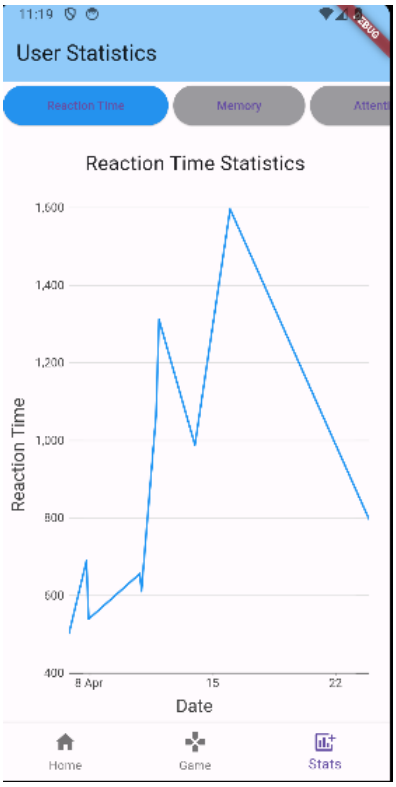


Figure 5.4: Statistics page.

The selection of categories is implemented using the `SingleChildScrollView` widget, which allows fitting buttons into a single row without exceeding the screen boundaries. For drawing the statistical graphs, the **_buildChart** widget is used, which returns a `charts.LocalDateTimeFactory`, allowing flexible customization of the graphs from choosing colors and setting axes to selecting the type of data for the graph, see Listing 5.4.

```
return charts.TimeSeriesChart(
  seriesList,
  animate: true,
  dateTimeFactory: const charts.LocalDateTimeFactory(),
  primaryMeasureAxis: charts.NumericAxisSpec(
    tickProviderSpec: charts.BasicNumericTickProviderSpec(zeroBound: false),
  ),
  domainAxis: charts.DateTimeAxisSpec(
    tickFormatterSpec: charts.AutoDateTimeTickFormatterSpec(
      day: charts.TimeFormatterSpec(format: ’d’, transitionFormat: ’d MMM’),
    ),
  ),
  behaviors: [
    charts.ChartTitle(
        ’Date’,
        behaviorPosition: charts.BehaviorPosition.bottom
    ),
    charts.ChartTitle(
        yAxisLabel,
        behaviorPosition: charts.BehaviorPosition.start
      ),
    ],
);
```

Listing 5.4: Setting charts configuration.

5.5.3 Navigation bar

Although the navigation bar is a small part of the application, it is no less important as it facilitates user navigation through the application's pages. To implement it, the `BottomNavigationBar` widget, which is part of the `<flutter/material.dart>` package, was used. The navigation bar consists of three items, each responsible for navigating to one of the following pages: the home page, the game page, and the statistics page.

This bar is present at the bottom of the screen on all pages except for the account creation and login pages. For this, navigation control using an index and the `bottomNavigationBar` widget has been added to the source code of the pages with the panel.

5.5.4 Account management

There are two pages in the application for managing accounts: one for creating a new account and another for logging into an existing one, see 5.1.1. Both utilize a user management service from Firebase, more in 4.2.2, called **Firebase Authentication** [4]. In addition to various SDKs, it also provides a UI for viewing and managing registered users. After connecting Firebase Authentication in the `Firebase Console`, the package `<firebase_auth/firebase_auth.dart>` is needed in the source code to use its features.

During account registration, the user will see a page with fields to enter their email, password, name, and age, (see Figure 5.5). To control the data that users enter in the fields, a TextEditingController is used, which allows managing text fields. For example, `_emailController` checks whether the email entered by the user matches the expected format. After clicking the **'Sign Up'** button, a new account is created using the method `createUserWithEmailAndPassword()`, which belongs to the `FirebaseAuth` class. Simultaneously, the user is automatically logged into the newly created account using the `signInWithEmailAndPassword()` method of the same class, and their data is added to Cloud Firestore using the `saveUserData()` function.

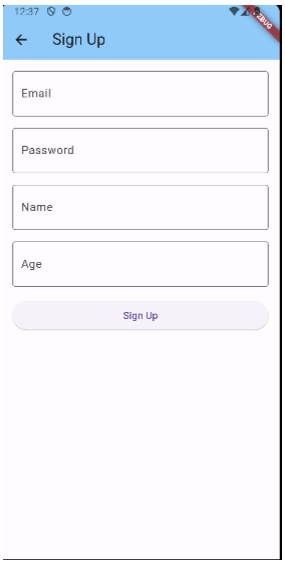


Figure 5.5: Sign up page.

When attempting to log into an account, the user needs to enter their email and password, which are also checked by controllers. Then, using the aforementioned `signInWithEmailAndPassword()` method, an attempt is made to log into the account. If the account does not exist, the user will be notified.

Chapter 6

Challenges and Testing

The testing of the application was carried out manually, not only by myself but also by a number of other users who had access to the app. The number of users that participated in testing the application was five. Based on the feedback received during the tests, adjustments were made to the design and functionality of the application. This chapter will discuss the challenges encountered, the tests conducted, and the changes that were implemented.

6.1 Homepage testing

During the testing of the main page of the application, which was discussed in 5.5.1, several inaccuracies with the leaderboard among players were discovered. The first issue was that user data was being retrieved incorrectly, resulting in the table always displaying only one player. Moreover, in cases where there is no network connection, the table, accordingly, cannot retrieve player data from the database.

The cause of the first error was due to a feature of Firestore, namely the ability for a collection to inherit from a document that is empty, which means it does not exist. In such cases, retrieving information stored in the collection is not possible. The solution was to add a “dummy„ field that serves as a placeholder, making it possible to extract information from a sub-collection, see 5.3.

The second inaccuracy was corrected using the `'connectivity_plus'` package. With its help, the presence of a network connection is checked on the user’s device, and in the absence of such a connection, they are notified.

6.2 Game testing

As a result of testing the game designed to assess human memory, attention, and reaction, which was detailed in 5.4, issues were discovered that indirectly affect not only the player’s experience but also the accuracy of the initial results.

A key error was a small delay in reaction time, which occurred because the timer in the source code started before the stimulus appeared on the player’s screen (see 5.4). While this delay was negligible, it was unacceptable in the context of measuring reaction time. The fix for the error was to subtract the delay from the result. To do this, another timer was started next to the reaction time timer, which counted down the time until a change appeared on the screen, and then the result of the second timer was subtracted from the first timer.

In the initial design of the game, the player's reaction time was displayed on the screen after each shown trio of squares. After test games, it was discovered that this label significantly distracted the player from the fields they needed to monitor, thereby worsening their results. Based on the aforementioned reason, this label was removed to simplify the game for the user and improve their results.

In addition, an error was discovered with the hover effect over fields upon clicking them, as it did not disappear but permanently changed the field to a different (gray) color. To correct this, the number of the field that the user clicked was added to an array which turned it gray, and then, after a delay of 200 milliseconds, the element was removed from the array and the field returned to its usual color.

6.3 Charts and statistics

Regarding the statistics page, which was described in 5.5.2, the tests revealed that the errors mostly concerned the frontend part. Although they were not complex to fix, they significantly impacted the user experience of the application.

One of these problems was the inconvenient display of data on the graphs: because every game of the user was plotted on the graph, it became overloaded with information, which reduced its readability for users. Therefore, it was decided to take the average value of the user's results for all games played during one hour. This not only improves the visual perception of the graph, but also prevents one badly played game from standing out among the others, thus simplifying the analysis of the player's progress.

Additionally, during the design of the page, a problem with category selection was discovered: the "attention„ category did not fit within the page boundaries, leading to the displacement of all content on the page. To resolve this situation, the `SingleChildScrollView` widget was used, which allows scrolling to access content that does not fit on the page (see 5.5.2).

6.4 Account registration

While using the application, an inconvenience was found in registering a new account (see 5.5.4). After successful account creation, the user was not automatically logged in to the account, which made it necessary to perform additional actions to log in to the newly created account. The function of automatic account login was added using the `signInWithEmailAndPassword()` function provided by the `Firebase Authentication` toolkit (see 5.5.4). With the addition of this part of the code, the user does not need to perform unnecessary actions, as now the login to the account is done automatically with its creation. At the same time, the user is automatically moved to the main page if there are no errors.

6.5 Survey

Among the players that used the application was conducted a survey to assess its overall functionality and not only. The survey was conducted using a questionnaire created in

Google Forms. The Table 6.1 shows the questions that were asked to users, most of them were in the form of a scale, but there were also some with answer options.

Table 6.1: Survey Results.

Questions	**Results**
Rate the game design (1 - did not like it, 5 - liked it)	“5„: 40% “4„: 40% “3„: 20%
How fun was the gaming process? (1 - boring, 5 - engaging)	“5„: 20% “4„: 40% “3„: 40%
How difficult was the game to understand? (1 - difficult, 5 - easy)	“5„: 20% “4„: 40% “3„: 20% “1„: 20%
How difficult was the game to play? (1 - difficult, 5 - easy)	“4„: 80% “3„: 20%
How do you like the visual part of the game? (1 - did not like it, 5 - liked it)	“5„: 20% “4„: 20% “3„: 40% “1„: 20%
How useful were charts with statistics?	“Useful„: 60% “Useless„: 20% “Did not see them„: 20%
How easy were charts to understand? (1 - complicated, 5 - easy)	“5„: 20%; “4„: 60%; “1„: 20%
Was the overall UI of the app comprehensible?	“Yes„: 80%; “No„: 20%
Would you play the game more in the future?	“Yes„: 60%; “No„: 40%
Would you recommend this game?	“Yes„: 80%; “No„: 20%

According to the results of the survey in the Table 6.1 the majority liked the gameplay and found it engaging. Also the complexity of the game most people considered as average, which indicates that the game fulfills its functions without complicating the process for the player. Some users had a hard time understanding what exactly to do to play the game, so the explanation of the game was amended in order to everyone could understand the steps to do in order to be successful in the game. Among the users who used the graphs with statistics, most of them found charts to be a useful and necessary part of the application to track the progress of the user. Also not less important is the general interface of the application, which according to the results of the survey was evaluated as easy to understand and user-friendly. At the end, some users shared their ideas for expanding the functionality of the game, namely adding more competition among users, for more fun and interest in the game.

Chapter 7

Conclusion

The purpose of this thesis was to develop and implement a game capable of assessing human memory, attention and reflexes simultaneously. During my work I studied the structure of the human brain, which parts of it are responsible for memory, attention and reaction, as well as ways to measure them. Also by analysing existing games to measure these human metrics, their main advantages and disadvantages were deduced, among which the main one was the lack of replayability and player engagement.

Based on the aforementioned data, I made several concepts of the game. The first of which was a game whose gameplay was aimed at maximum concentration of the player, as it involved different actions that should be done during the game: remember the correct order of squares, watch for the updating of the field in time by clicking on it, as well as filter the fields of other colours. Next was a game that tested memory, attention and reflexes with the help of cards with information that went sequentially. Based on them, questions would be created for the user to answer. Although the strength of this concept is a large set of context that would have made the game interesting, but since it does not provide an opportunity to accurately measure attention and reaction time, as well as cognitive tasks in the questions would have made the process very complicated, which could have affected the results. The last concept was a game based on Stroop's concept. The main disadvantage, because of which it was not chosen, is the excessively high complexity of the gameplay, which together with the reflex testing would have overloaded the user, thus not giving accurate results.

The next step was to define the software stack that would be used to implement the game chosen earlier. The Flutter framework was chosen as the cornerstone on which the application was built for its high performance, and the application itself was written in the Dart language. For the database was used NoSQL database Cloud Firestore, which is part of the toolkit Firebase. The Firebase Authentication function was also used from this toolkit.

During the implementation was realised not only the functionality of the game, but also additional elements of the application, such as: graphs with statistics for easy viewing of the progress of the user, the table of leaders, to raise interest in the game and also the ability to manage the account within the application. The final stage was testing among the users, after which on the basis of their feedback were made some corrections in the functioning of the game, as well as was made a survey among users to find out the overall assessment of the application.

The end result of the implemented application fully met the requirements of this thesis: a game with a clear and user-friendly user interface, as well as an exciting design capable of simultaneously measuring the player's memory, attention and reflexes. Compared to other solutions that are customised for one specific metric, my design allows the user to test all three simultaneously without losing quality.

7.1 Implementation improvements

Since functionally the game fulfills the tasks set for it, from the possible extensions of the application could be pointed out the expansion of the competitive functionality among users, where they will compete for superiority, by adding points that a player can get after the game depending on his performance. Also adding the ability for the user to view the successes of other players will add interest and fascination to the gameplay.

Bibliography

[1] CogniFit. *Visual episodic memory test* [online]. [cit. 2024-04-17]. Available at: https://www.cognifit.com/aplicaciones/html5/public/assessment/ASSESSMENT~@~VISUAL_EPISODIC?testButtonUrl=https://www.cognifit.com/memory-test%3Freg=true%26regut=normal%26irclickid=VzqWpBX0XxyNR4K0-Lw1QVHwUkAQs7V9Xx80zQ0%26irgwc=1.

[2] Flutter. *Flutter architectural overview* [online]. [cit. 2024-05-01]. Available at: https://docs.flutter.dev/resources/architectural-overview.

[3] Flutter. *StatefulWidget class* [online]. [cit. 2024-05-01]. Available at: https://api.flutter.dev/flutter/widgets/StatefulWidget-class.html.

[4] Google. *Firebase Authentication* [online]. 2024 [cit. 2024-04-28]. Available at: https://firebase.google.com/docs/auth.

[5] Heaning, E. *Henry Gustav Molaison: The Curious Case of Patient H.M.* [online]. 2023 [cit. 2024-04-14]. Available at: https://www.simplypsychology.org/henry-molaison-patient-hm.html.

[6] Human Benchmark. *Reaction time test* [online]. [cit. 2024-04-17]. Available at: https://humanbenchmark.com/tests/reactiontime.

[7] Justpark. *Reaction time test* [online]. [cit. 2024-04-17]. Available at: https://www.justpark.com/creative/reaction-time-test/.

[8] Kendra Cherry, MSEd. *How Psychologists Define Attention* [online]. 2022 [cit. 2024-05-01]. Available at: https://www.verywellmind.com/what-is-attention-2795009.

[9] Larocco, J., Tahmina, Q., Lecian, S., Moore, J., Helbig, C. et al. Evaluation of an English language phoneme-based imagined speech brain computer interface with low-cost electroencephalography. *Frontiers in Neuroinformatics* [online]. 2023, vol. 17, [cit. 2024-05-07]. Available at: https://www.researchgate.net/figure/Electrocephalographic-headset-used-for-data-acquisition-shown-with-16-electrodes-in_fig1_376630877.

[10] Lebrun, J.-L. *An Introduction to the Event-Related Potential Technique.* 2nd ed. The MIT Press, 2014. ISBN 9780262525855.

[11] Lesley University. *What the Stroop Effect Reveals About Our Minds* [online]. [cit. 2024-04-28]. Available at: https://lesley.edu/article/what-the-stroop-effect-reveals-about-our-minds.

[12] Marco Cascella, Y. A. K. *Short-Term Memory Impairment* [online]. 2023 [cit. 2024-04-14]. Available at: https://www.ncbi.nlm.nih.gov/books/NBK545136/.

[13] OpenStaxCollege. *Parts of the Brain Involved with Memory* [online]. OpenStaxCollege [cit. 2024-05-07]. Available at: https://pressbooks-dev.oer.hawaii.edu/psychology/chapter/parts-of-the-brain-involved-with-memory/#:~:text=The%20main%20parts%20of%20the,as%20well%20as%20recognition%20memory.

[14] Richardson, J. T. Knox's cube imitation test: A historical review and an experimental analysis. *Brain and Cognition*. 2005, vol. 59, no. 2, p. 183–213. ISSN 0278-2626. Available at: https://www.sciencedirect.com/science/article/pii/S0278262605000941.

[15] Salzman, C. D. *Amygdala* [online]. 2024 [cit. 2024-04-14]. Available at: https://www.britannica.com/science/amygdala.

[16] Schraven, T. *The Hipp Chronoscope*. [cit. 2024-05-01].

[17] Smallcombe, M. *SQL vs NoSQL: 5 Critical Differences* [online]. 2024 [cit. 2024-05-01]. Available at: https://www.integrate.io/blog/the-sql-vs-nosql-difference/.

[18] Soma Technologies. *Understanding Reaction Time and its Components* [online]. 2023 [cit. 2024-05-07]. Available at: https://www.linkedin.com/pulse/defining-reaction-time-understanding-its-components/.

[19] Washington.edu. *Short term memory test* [online]. [cit. 2024-04-17]. Available at: http://faculty.washington.edu/chudler/stm0.html.

[20] Wikipedia contributors. *Depth-first search* [online]. Wikipedia, The Free Encyclopedia, 2024 [cit. 2024-04-17]. Available at: https://en.wikipedia.org/wiki/Depth-first_search.

[21] Wikipedia contributors. *Event-related potential* [online]. Wikipedia, The Free Encyclopedia, 2024 [cit. 2024-04-17]. Available at: https://en.wikipedia.org/w/index.php?title=Event-related_potential&oldid=1214912714.

[22] Wikipedia contributors. *Test of Variables of Attention* [online]. Wikipedia, The Free Encyclopedia, 2024 [cit. 2024-04-17]. Available at: https://en.wikipedia.org/w/index.php?title=Event-related_potential&oldid=1214912714.

[23] Yassa, M. A. *Hippocampus* [online]. 2024 [cit. 2024-04-14]. Available at: https://www.britannica.com/science/hippocampus.

Appendix A

Contents of the attached medium

- `/color_sequence_game` – source code of the application.
- `/latex` – source code of the thesis.
- `xpetro27.pdf` – text of the thesis.
- `README.md` – documentation of the application.