



POLITECHNIKA ŚLĄSKA
WYDZIAŁ INŻYNIERII BIOMEDYCZNEJ

Projekt inżynierski

Aplikacja do testowania pamięci krótkotrwałej

Autor: Kamil Olesz

Kierujący pracą: Dr inż. Jacek Kawa

Zabrze, 28 grudnia 2017

Spis treści

1. Wstęp	1
2. Projekt aplikacji	3
2.1 Założenia wstępne	3
2.2 Opcjonalne funkcje	3
2.3 Metodologia	4
3. Specyfikacja wewnętrzna	5
3.1 Wprowadzenie	5
3.2 Interfejs użytkownika	6
3.2.1 activity_menu	6
3.2.2 activity_numbers	7
3.2.3 activity_options	8
3.2.4 activity_results	8
3.2.5 list_row	9
3.2.6 frame	9
3.2.7 gradient	9
3.3 Klasy i metody	10
3.3.1 Menu	10
3.3.2 Numbers	10
3.3.3 Tutorial	12
3.3.4 Options	12
3.3.5 Results	12
3.3.6 MultirowLVAdapter	12
4. Specyfikacja zewnętrzna	13
4.1 Menu główne	13
4.2 Rozgrywka	13
4.3 Samouczek	13
4.4 Opcje	13
4.5 Wyniki	13
5. Testy	15
6. Podsumowanie i wnioski	17

<i>Bibliografia</i>	18
-------------------------------	----

1. Wstęp

W dzisiejszych czasach nauka rozwija się w zawrotnym tempie. Mimo tego, zagadnienie działania ludzkiego mózgu wciąż pozostaje nie do końca rozwiązane [3]. Jedną z fundamentalnych umiejętności mózgu, bez której człowiek nie mógłby funkcjonować, jest pamięć. Odpowiada ona za wszystkie zdolności, których nauczył się człowiek przez całe swoje życie. Bez pamięci, wszystkie wrażenia, które ludzie doświadczają, zacierają się i to, co człowiek odczuł, po chwili stałoby się dla niego nieistniejące [3, 4].

Za zdolność zapamiętywania odpowiedzialne są neurony, a dokładnie ich całe sieci połączeń. Przy doświadczeniu jakiegokolwiek bodźca, komórki nerwowe łączą się ze sobą. Takie połączenie zwane jest synapsą – może być mocniejsze lub słabsze, co wpływa na późniejsze utrzymanie tegoż wiązania. Po każdorazowym odebraniu tego samego wrażenia, synapsa wzmacnia się i pozostaje nieprzerwana. Przy jednorazowym, krótkim odebraniu bodźca, synapsa szybko zanika gdyż połączenie między komórkami jest zbyt słabe [5, 7]. Mózg sam potrafi ustalić ważność pobudzenia, zatem wybiórczo pozostawia rzeczy w swojej pamięci. Tak dla przykładu, pierwsze poparzenie się ogniem „uczy” człowieka, by ten starał się unikać kolejnych poparzeń. Wrażenie bólu jest bowiem ważnym aspektem dla mózgu. Z drugiej strony, mało ważną informacją dla człowieka jest wygląd mijającej go na chodniku osoby. Mózg przyswaja ten wygląd, lecz po chwili może dojść do wniosku, że nie jest to istotna informacja i połączenie neuronów się zerwie. Wszystkie operacje wiązań dzieją się w strukturze mózgu zwanej hipokampem. [1, 2].

Najbardziej popularnym podziałem pamięci jest podział ze względu na czas trwania pamiętania – ultrakrótkotrwała (zwana również sensoryczną), krótkotrwała oraz długotrwała. W dalszej części pracy uwaga zostanie skupiona tylko na pamięci krótkotrwałej [4].

Pamięć krótkotrwała jest wykorzystywana przez ludzi codziennie, przy zwykłych domowych czynnościach, do których nie przywiązuje się zbyt dużej wagi, np. przy zapamiętaniu czy herbata została już posłodzona albo drzwi wejściowe zostały zamknięte. Warto ćwiczyć tę umiejętność, by sprawniej wykonywać codzienne czynności (również te w pracy) oraz lepiej przyswajać informacje świata nas otaczającego. Trening pamięci krótkotrwałej ma również wpływ na zdolność pamiętania długoterminowego. Więcej neuronów jest wykorzystywanych w obrębie ośrodka pamięci, przez co człowiek jest w stanie zapamiętać więcej rzeczy. Pamiętając pewne informacje dłużej, łatwiej jest przenieść je z obszaru pamięci krótkotrwałej na długotrwałą [3–5].

Istnieje wiele sposobów testowania ludzkiej pamięci, a najczęściej odnosi się do badania pamięci krótkotrwałej. Przeprowadzenie takiego testu wiąże się z prostym zadaniem zapamiętania pewnych rzeczy, np. figur geometrycznych, par wyrazów, krótkich słów

czy ciągów cyfr. Prezentowana w pracy aplikacja, będzie odnosiła się do ostatniego testu [6].

Trening pamięci polega na regularnym powtarzaniu danych testów. Głównie warto skupić się na jednym, gdyż powtarzane czynności są bardziej przyswajane przez mózg. Powtarzanie ciągu cyfr jest uniwersalne dla każdej osoby, w końcu każdy zna cyfry, a bardzo łatwo przy tym można określić jak długi ciąg zapamiętuje dana osoba [4, 6].

Jest mnóstwo form trenowania (przy tym i testowania) pamięci [6]. Najbardziej odpowiednią w dzisiejszych czasach są wszelkie aplikacje mobilne. Na sam trening pamięci konieczna jest tylko chwila czasu, ale najważniejsza jest regularność, bowiem efekty widoczne są dopiero po długim okresie. Łatwo jest włączyć aplikację mobilną, korzystać z niej przez określony czas po czym bez żadnego opóźnienia wrócić do codziennych obowiązków. Treningi takie można wykonywać w każdych wolnych chwilach, chociażby przy podróży autobusem. To daje ogromną wygodę i przez to, coraz więcej ludzi decyduje się na trenowanie swojej pamięci – dlatego też występuje popyt na tego rodzaju oprogramowania.

Dostępne na rynku aplikacje ^{1 2 3} oferują przerost formy nad treścią. Twórcy skupiają uwagę na przyciągnięciu potencjalnych klientów swoim wyglądem. Sama forma aplikacji może w znaczący sposób dekoncentrować trenującego (np. zmiana kolorów ekranu, zbyt długie animacje czy migające obrazki). Każdy oczekuje od aplikacji czegoś innego, tak więc przydatna byłaby możliwość personalizacji, lub ewentualnie możliwość łatwej edycji programu i sprzedaż spersonalizowanych form.

Celem opisanej pracy jest zaprojektowanie i stworzenie aplikacji mobilnej, umożliwiającej prosty test pamięci krótkotrwałej. Dodatkowo, Aplikacja powinna mieć formę pozwalającą na trening tego rodzaju pamięci. Program ma być łatwy w korzystaniu przez każdego użytkownika oraz spełniać jego potrzeby. Docelowe grupy osób to:

- osoby zdrowe, nie trenujące dotychczas pamięci,
- osoby zdrowe, trenujące regularnie pamięć krótkotrwałą,
- osoby chore na wszelkie ubytki pamięci – korzystanie w celach diagnostycznych i terapeutycznych,
- osoby młodsze – forma na tyle prosta by mogła korzystać z niej jak najmłodsza osoba
- osoby starsze – podobnie jak dla osób młodszych, forma na tyle prosta by bez problemów mogły korzystać z aplikacji starsze osoby.

Realizacja celu wymagać będzie następujących kroków: projekt, implementacja i testowanie. Zostaną one opisane w dalszej części pracy.

¹ Brainwell — Brain Training Games By Monclarity, LLC – App Store, 17.12.2017

² Lumosity – Trening mózgu, Lumos Labs, Inc. – Google Play, 17.12.2017

³ Gry na pamięć: Trening mózgu, Maple Media – Google Play, 17.12.2017

2. Projekt aplikacji

2.1 Założenia wstępne

Program musi zawierać prosty test pamięci krótkotrwałej, jakim jest zapamiętanie losowo generowanego ciągu cyfr. Test musi zostać zaimplementowany w aplikacji na urządzenia mobilne. Da to możliwości sprawdzenia pamięci w każdej chwili praktycznie każdej osobie. Dodatkowo, aplikacja powinna mieć regulowaną długość ciągu, co umożliwi test pamięci według umiejętności użytkownika.

Aplikacja ma być stworzona w formie programu umożliwiającego również trening pamięci. Dlatego poziom trudności powinien rosnąć wraz z umiejętnościami użytkownika. Powinien istnieć również warunek zakończenia testu czy treningu. Nie powinno być żadnego ograniczenia w związku z maksymalną czy minimalną długością ciągu. W końcu ktoś może mieć trudności z zapamiętaniem już jednej cyfry. Z drugiej strony, użytkownik może mieć ponadprzeciętną pamięć a zadanie maksymalnej długości ciągu może spowodować, że nie będzie sensu dalszego treningu pamięci, ponieważ bez większego wysiłku umysłowego byłby w stanie zapamiętać maksymalnie długi ciąg. Oczywiście muszą pojawiać się komunikaty informujące użytkownika o poprawnym lub błędnym zapamiętaniu wygenerowanego ciągu.

2.2 Opcjonalne funkcje

Dodatkową funkcją oprogramowania może być wprowadzanie głosowe zapamiętanego ciągu. Umożliwi to korzystanie z aplikacji osobom starszym, dla których problemem byłoby szybkie wpisanie z klawiatury numerycznej. Kolejnym plusem takiego rozwiązania, byłoby umożliwienie testowania pamięci krótkotrwałej osobom z dysfunkcjami ruchowymi, np. po udarze. Opcja wprowadzania głosowego byłaby również formą personalizacji. Dla pewnych osób wygodniejsze mogłoby być wypowiedzenie zapamiętanego ciągu niż wpisanie go.

Kolejną opcjonalnością mogłoby być zapamiętanie danych wyników. Z racji treningu pamięci, przydatna byłaby możliwość sprawdzenia i porównania jak długi ciąg użytkownik zapamiętał miesiąc temu, a jakie wyniki osiąga teraz. Zapisywanie wyników mogłoby odbywać się w formie zapisu do bazy danych, jednak wiązałoby się to z koniecznością dostępu do internetu podczas korzystania z aplikacji. Alternatywą tego rozwiązania byłby zapis wyników do pamięci urządzenia mobilnego. Wiazałoby się to jednak z koniecznością wysłania wyników do dogłębnej analizy chociażby przez lekarzy.

2.3 Metodologia

W celu wypełnienia wymienionych założeń, aplikacja zostanie zaprogramowana w języku Java w środowisku Android Studio. Umożliwia to stworzenie aplikacji na urządzenia mobilne z systemem android, czyli najbardziej popularnym systemem wśród obywateli Polski¹.

Aplikacja będzie posiadała ustawienia, umożliwiające wybranie początkowej długości ciągu, zadeklarowanie liczby szans – warunku kończącego rozgrywkę oraz wybór trybu wpisywania głosowego (zamień tekst lub dodaj do tekstu). Kolejnym dodatkowym aspektem będzie zapisywanie wyniku najdłuższego zapamiętanego ciągu cyfr w danej rozgrywce wraz z zapisem czasu zapamiętywania tej sekwencji.

Dla przejrzystości rozgrywki, program będzie posiadał samouczek, dostępny do uruchomienia z pozycji menu głównego. Wprowadzi on w panujące w trakcie testu zasady – co należy zrobić w danym etapie rozgrywki. Zaznajomi również użytkownika z funkcją wprowadzania głosowego.

Test pamięci będzie przebiegał według następującego schematu:

1. Rozpoczęcie testu po uprzednim ustawieniu początkowej długości ciągu.
2. Wygenerowanie losowego ciągu cyfr od 0 do 9.
3. Zapamiętanie wyświetlonego ciągu przez użytkownika.
4. Wprowadzenie zapamiętanej sekwencji.
5. Sprawdzenie poprawności zapamiętania.
6. Jeżeli warunek końca testu nie został spełniony - przejście do punktu drugiego.
7. Jeżeli warunek został spełniony - zakończenie rozgrywki.

¹ Według statcounter GlobalStats, 18.12.2017

3. Specyfikacja wewnętrzna

3.1 Wprowadzenie

Aplikacja została napisana w środowisku Android Studio. Klasy są napisane w języku Java a pliki layoutu mają rozszerzenie xml. Minimalna wspierana wersja API to 16, co odpowiada wersji 4.1 Jelly Bean systemu operacyjnego Android. Według Android Studio, daje to możliwość uruchomienia stworzonej aplikacji na, w przybliżeniu, 99.2% urządzeń aktywnych na Google Play Store, co daje możliwość testowania pamięci na praktycznie każdym urządzeniu mobilnym z systemem Android.

Aplikacja korzysta z jednej zewnętrznej biblioteki, a mianowicie biblioteki opencsv, pozwalającej na zapis i odczyt danych do/z plików o rozszerzeniu CSV (comma-separated values). Reszta wymaganych bibliotek dostarcza samo środowisko. Biblioteka opencsv jest biblioteką typ open source, możliwą do pobrania na stronie opencsv.sourceforge.net. Przy programowaniu zostały użyte następujące klasy i metody tej biblioteki:

- CSVWriter - klasa umożliwiająca zapisanie danych tablicowych do pliku typu CSV. Jako argumenty obiektu tej klasy podawany jest obiekt klasy FileWriter (przyjmuje ścieżkę oraz nazwę pliku, służy do ogólnego zapisywania plików) oraz znak separujący. Wykorzystywane metody to:
 1. Metoda writeNext, zapisująca kolejny wers do pliku. Wers jest podawany w formie tablicy z danymi typu String.
 2. Metoda close, zapisująca i zamykająca utworzony lub modyfikowany plik.
- CSVReader - klasa umożliwiająca odczyt z pliku CSV. Jako argumenty obiektu tej klasy podawany jest obiekt klasy FileWriter oraz znak separujący. Wykorzystywane metody to:
 1. Metoda readAll, odczytująca wszystkie wersy z danego pliku CSV. Zapisuje każdą komórkę do odpowiedniego miejsca w liście typu String tablicowy.
 2. Metoda close, zamykająca otwarty wcześniej plik CSV.

3.2 Interfejs użytkownika

Interfejs graficzny użytkownika tworzą następujące pliki:

- activity_menu.xml,
- activity_numbers.xml,
- activity_options.xml,
- activity_results.xml,
- list_row.xml,
- frame.xml,
- gradient.xml.

3.2.1 activity_menu

Jest to widok ukazywany od razu przy uruchomieniu aplikacji (Rys. 3.1). Składa się z czterech przycisków i jednego pola tekstowego. Każdy przycisk odpowiedzialny jest za przejście do innej aktywności, ukazanej w nazwie przycisku. Pole tekstowe informuje użytkownika, że znajduje się w menu głównym.



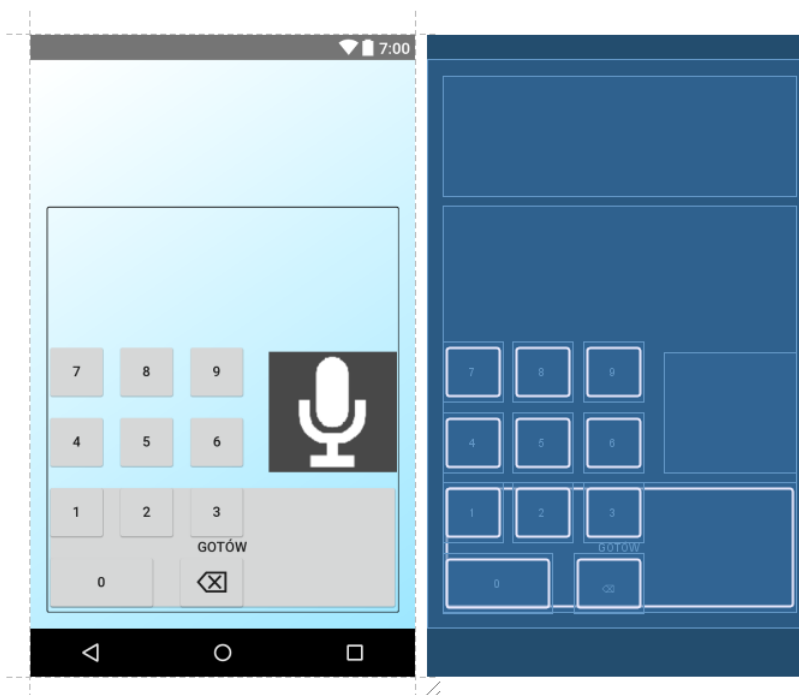
Rys. 3.1: Projekt wyglądu menu głównego aplikacji.

3.2.2 activity_numbers

Widok pokazujący wygląd całego testu pamięci (Rys. 3.2). Zawiera 10 przycisków odpowiedzialnych za wpisywanie cyfr. Są ułożone w wygodny do użytkowania sposób. Przyciski te są oznaczone odpowiednim symbolem, przez co użytkownik wie, jaka cyfra pojawi się po naciśnięciu na daną kontrolkę. Do usuwania wpisywanego przez omówione elementy ciągu, służy przycisk z wizerunkiem standardowego klawisza do usuwania tekstu w klawiaturach systemu Android. Kolejną kontrolką jest przycisk z napisem „GOTÓW”. Służy on do wyrażenia gotowości i przejścia do następnego etapu rozgrywki. ImageButton, z wizerunkiem mikrofonu, służy do włączenia wprowadzania głosowego.

Widok zawiera dwa pola tekstowe. Pierwsze, na samej górze, służy do informowania użytkownika o aktualnym etapie testu. W drugim, poniżej, ukazuje się ciąg do zapamiętania, wpisywana przez użytkownika zapamiętana sekwencja oraz statystyki po zakończeniu rozgrywki. Dodatkowo, dookoła drugiego pola tekstowego znajduje się obramowanie. Wygląd ramki został utworzony w pliku frame.xml opisanym w rozdziale 3.2.6.

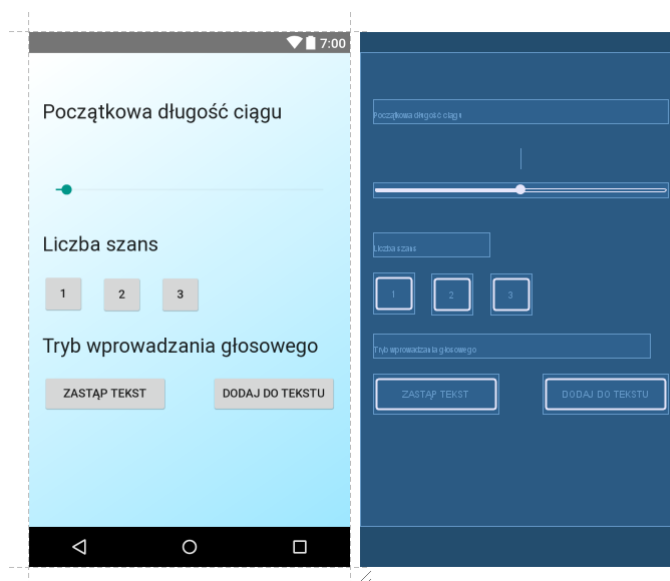
W projekcie widoku niektóre elementy nakładają się na siebie, jednakże podczas testu, kontrolki zostały tak zaprogramowane, że zmieniają swoje położenie w zależności od danego etapu rozgrywki. Wszystko jest zaprojektowane w taki sposób, by było jak najbardziej czytelne i intuicyjne dla użytkownika.



Rys. 3.2: Projekt wyglądu okna rozgrywki.

3.2.3 activity_options

Okno opcji pokazuje dostępną możliwość konfiguracji (Rys. 3.3). Trzy pola tekstowe informują użytkownika jakie ustawienia modyfikuje. Znajdujące się przyciski pokazują swoim tekstem skutek zmiany danej opcji. Przesunięcie suwaka zmienia początkową długość ciągu. Jest on przesuwalny w wartościach od 0 do 49, jednak do wybranej wartości suwaka dodawana jest jedynka. Da to możliwość ustawienia początkowej długości ciągu w zakresie od 1 do 50. Pole tekstowe, znajdujące się nad suwakiem, wyświetla zadaną wartość początkowej długości sekwencji.

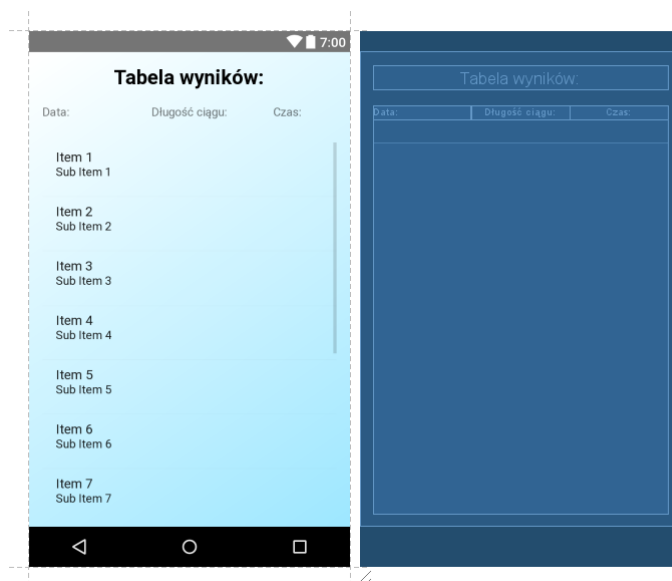


Rys. 3.3: Projekt wyglądu menu opcji.

3.2.4 activity_results

Jest to plik widoku, odpowiedzialny za wyświetlenie wyników ukończonych testów. Informuje o tym pole tekstowe, znajdujące się na samej górze okna. Poniżej znajduje się liniowy układ horyzontalny, w skład którego wchodzi trzy pola tekstowe. Informują one o typie wyświetlanej pod tekstem pozycji.

W celu wyświetlenia samych wyników, została dodana kontrolka ListView. Lista jest przewijalna i nieograniczona w ilości pokazywanych elementów. W celu prawidłowego wyświetlania żądanych wyników, został dodany plik formatujący pojedynczy wiersz listy. Został on opisany w rozdziale 3.2.5.



Rys. 3.4: Projekt wyglądu listy wyników.

3.2.5 list_row

Układ ustawiający wygląd pojedynczego wierszu listy z wynikami. Jeden wiersz składa się z trzech pól – data wykonania testu, długość zapamiętanego ciągu oraz czas zapamiętywania najdłuższej, poprawnie odtworzonej sekwencji. Pole z datą jest odrobinę dłuższe od pozostałych pól. Zapobiega to zawijaniu się tekstu w danym wierszu.

3.2.6 frame

Plik zawiera styl ramki wokół pola tekstowego wyświetlającego np. ciąg cyfr do zapamiętania. Ramka jest koloru czarnego, ma lekko zaokrąglone rogi, a jej szerokość wynosi 1 dp (density independent pixel). Tło wnętrza ramki zostało ustawione na przezroczyste, aby nie zasłaniało tła całego okna.

3.2.7 gradient

Odpowiedzialny jest za tło każdego okna aplikacji. Środowisko Android Studio umożliwia w łatwy sposób utworzenie gradientu o zadanych parametrach. Tak wygenerowany gradient należy załączyć do każdego pliku z widokiem, gdzie przeskaluje się na całe okno lub inny żądany wymiar.

3.3 Klasy i metody

Kod źródłowy aplikacji składa się z następujących klas:

- Menu.java,
- Numbers.java,
- Tutorial.java,
- Options.java,
- Results.java,
- MultirowLVAdapter.java.

Ważnym plikiem jest również plik `AndroidManifest.xml`, w którym zawarte są pozwolenia na zapis i odczyt danych z pamięci urządzenia. Plik ten zawiera również ustawienie początkowej aktywności, uruchamianej zaraz po załączeniu aplikacji. Dodatkowo określona jest orientacja ekranu każdego z widoków na „portrait”, co odpowiada pionowemu ułożeniu telefonu.

3.3.1 Menu

Klasa obsługująca akcje wykonywane w menu głównym aplikacji. Pobiera kontrolki z widoku `activity_menu` (Opisany w rozdziale 3.2.1). Do przycisków przyporządkowana jest akcja rozpoczęcia odpowiadającej przyciskowi aktywności. Tak dla przykładu, po naciśnięciu klawiszu z napisem „Samouczek”, poprzez metodę `startActivity` (metoda środowiska Android Studio), zostaje uruchomiona aktywność odpowiedzialna za samouczek.

W tej aktywności, poprzez metodę `verifyStoragePermissions`, wysyłane jest żądanie o udzielenie pozwolenia na zapis i odczyt plików w pamięci urządzenia. Na początku sprawdzane jest czy aplikacja już nie ma przyznanego pozwolenia. Jeżeli nie ma, wysyłane jest zapytanie, a jeśli zezwolenie zostało przyznane, to nic nie robi. Taki zabieg umożliwia pominięcie wysłania żądania przy każdym uruchomieniu aplikacji, a nawet przy każdym wejściu do menu głównego.

Mimo, że przyzwolenia na zapis i odczyt są już przyznane w pliku `AndroidManifest.xml`, to dla nowszych wersji systemu Android (Android 6.0+), wymagane jest wyświetlenie tej informacji użytkownikowi i zapytanie go o wyrażoną zgodę.

3.3.2 Numbers

Klasa obsługująca cały przebieg testu pamięci krótkotrwałej. Pobiera on widok z pliku `activity_numbers.xml` (Opisany w rozdziale 3.2.2). Głównym elementem działania programu jest metoda `onCreate`, a dokładniej nasłuchiwanie na naciśnięcie przycisku z tagiem 'ok'. Po jego naciśnięciu, w zależności od etapu testu, uruchamiane są odpowiednie metody. Aktualny etap rozgrywki zależny jest od wartości zmiennej sterującej

'steering'. Zmienna ta przyjmuje wartości 0, 1 lub 2. Pierwsza wartość określa etap zapamiętywania ciągu cyfr przez użytkownika. W tym etapie uruchamia się pomiar czasu zapamiętywania. Generuje się również ciąg cyfr, po czym wyświetlany jest na ekranie. Przy naciśnięciu na przycisk wyrażający gotowość, pomiar czasu zapamiętywania zatrzymuje się, zmienia się wygląd okna aplikacji, oraz zmienna sterująca przyjmuje wartość 1. Ta wartość określa etap wpisywania zapamiętanej sekwencji. Po kolejnym przyciśnięciu, w zależności od tego, czy użytkownikowi zostały jeszcze szanse lub nie, zmienna sterująca przyjmuje wartości odpowiednio 0 lub 2. Wartość 2 określa zakończenie testu, gdzie po ponownym naciśnięciu na omawiany przycisk, aplikacja „wraca” do menu głównego. Przy tym etapie rozgrywki wyświetlana jest informacja o końcu gry oraz ukazywane są statystyki – najdłuższy zapamiętany ciąg oraz czas jego zapamiętywania. Jeśli użytkownikowi nie udało się zapamiętać żadnej sekwencji, Aplikacja wychwyci to i poinformuje użytkownika o tym wyjątku.

Wygenerowanie losowego ciągu cyfr przebiega w metodzie `random_number`. Przyjmuje ona parametr `'length'`, który odpowiada długości wylosowanej sekwencji. Zadeklarowana zostaje zmienna tablicowa typu `int` o rozmiarze równym wartości przesłanego parametru. Następnie, do każdej komórki tejże tablicy zostaje wpisana losowa cyfra. Tak wygenerowany ciąg zostaje zwrócony do dalszych działań.

Kolejnym krokiem jest przekształcenie wygenerowanego ciągu z formy tablicy liczb całkowitych do typu `String`. Dzieje się to w metodzie `intarray2string`, która przyjmuje parametr typu `int` tablicowy. Pętla przechodzi po każdej komórce tablicy i dopisuje jej wartość do napisu. Przekonwertowany ciąg jest zwracany do dalszego użytku. Zabieg ten jest konieczny przy bezproblemowym wyświetleniu ciągu w polu tekstowym oraz łatwym porównaniu wartości wpisanej przez użytkownika z zapamiętywaną sekwencją.

Metoda sprawdzająca poprawność zapamiętanej sekwencji to metoda `check`. Przyjmuje ona dwie zmienne typu `String`. Jedna zawiera ciąg cyfr do zapamiętania, a druga wartość wpisaną przez użytkownika po zapamiętaniu. Funkcja sprawdza czy napisy są sobie zbieżne. Jeżeli tak, użytkownikowi zostaje wyświetlony stosowny komunikat. Długość sekwencji zostaje zwiększona o jeden, czas zapamiętywania zostaje zapisany a także zmienna `'remembered'` zostaje ustawiona na `true`. Zmienna ta jest typu `bool` i zawiera informacje o tym, czy użytkownik zapamiętał jakikolwiek ciąg. Do tego etapu, zmienna ta jest ustawiona na `false`. Przy negatywnym wyniku porównania, zmniejszana jest liczba szans. Jeżeli liczba szans jest większa od zera, użytkownikowi wyświetlony jest komunikat o pozostałej liczbie szans. W przeciwnym przypadku, gracz zostaje poinformowany o zakończeniu rozgrywki.

Metoda odpowiedzialna za zmianę wyglądu aplikacji w zależności od aktualnego etapu testu pamięci, to metoda `invisibility.....`

3.3.3 Tutorial**3.3.4 Options****3.3.5 Results****3.3.6 MultirowLVAdapter**

4. Specyfikacja zewnętrzna

4.1 Menu główne

4.2 Rozgrywka

4.3 Samouczek

4.4 Opcje

4.5 Wyniki

5. Testy

Tutaj właśnie efekt przetestowania aplikacji na grupie osób: czy coś z tego wynika (od opinii po wyniki numeryczne, zależnie od tego, co uda się zrobić)

+ ewentualne uwagi: co warto rozwijać itp.

6. Podsumowanie i wnioski

Celem pracy było Zrobiono to, tamto i siamto. Cel pracy został osiągnięty (lub nie)

Bibliografia

- [1] ANDERSON, J. R. *Uczenie się i pamięć – integracja zagadnień*. Wydawnictwa Szkolne i Pedagogiczne, Warszawa, 1998.
- [2] BUDOHOSKA, W., W. Z. *Psychologia uczenia się*. Przegląd badań eksperymentalnych i teorii. Państwowe Wydawnictwo Naukowe, Warszawa, 1977.
- [3] GAZZANIGA, M. S. *The Cognitive Neurosciences*, 4 ed. The MIT Press, Londyn, 2009.
- [4] JAGODZIŃSKA, M. *Psychologia pamięci. Badania, teorie, zastosowania*. Sensus, 2008.
- [5] JODZIO, K. *Neuronalny Świat umysłu*. Oficyna wydawnicza Impuls, Kraków, 2005.
- [6] NIEDŹWIEŃSKA, A. Rodzaje testów do badania pamięci.
- [7] VETULANI, J. Pamięć: podstawy neurobiologiczne i możliwości wspomagania.