Politechnika Poznańska
Wydział Automatyki, Robotyki i Elektrotechniki

Szymon Kwiatkowski

13 czerwca 2022

Aplikacje Mobilne

Sudoku

Spis treści

1. Wykorzystane elementy

- Podstawowe elementy graficzne(button etc.) - 5%
- Średnie elementy(fragment itp.) - 10%
- Dodatkowe UI(progress bar, własna klasa view i rysowanie canvas, customowy clock, grid layout, imagebutton) 30%
- Wielojęzyczność 5%
- Zgodność MVVC(dla klas view które posiadają dane: FragmentGame oraz FragmentScore) - 10%
- Biblioteki zewnętrzne(Live Data, Data Binding, Gson, Couritines, ViewModel, Canvas, Safe Args) - 30%
- Wielowątkowość - 5%

Sumarycznie zdobyto 95% punktów.

2. Interfejs aplikacji

2.1. Podział na fragmenty i nawigacja

Aplikacja składa się z 4 głównych fragmentów i są nimi:

- FragmentEntry - Fragment który jest domyślnym miejscem docelowym aplikacji po jej uruchomieniu. W tym menu możemy wybrać wyjście z aplikacji oraz przycisk start którym przechodzimy do kolejnego fragmentu w nawigacji.
- FragmentStart - w tym fragmencie użytkownik posiada 3 przyciski które pozwalają mu na wybór poziomu trudności gry. Po ich wybraniu użytkownik przechodzi do kolejnego etapu.
- FragmentGame - w tym fragmencie gracz ma możliwość ukończenia gry w sudoku. Liczby są wpisywane za pomocą zaznaczenia odpowiedniego pola i następnie wybrania liczby za pomocą przycisków w dolnej części ekranu które wpiszą daną liczbę w pole. Dodatkowo użytkownik posiada przycisk pozwalający na tworzenie notatek w danym zaznaczonym polu. Oprócz tego interfejs zwraca informacje o zdobytych punktach które otrzymujemy poprzez poprawny wybór liczby w dane pole. Ilość punktów zależna jest od wybranego poziomu trudności. Dodatkowo użytkownik posiada informacje o ilości wypełnionych pól za pomocą ProgressBar oraz informację o czasie rozwiązywania sudoku wyświetlanym w polu tekstowym. Po wypełnieniu wszystkich pól liczbowych użytkownik jest domyślnie przenoszony do kolejnego fragmentu.

- FragmentScore - jest to fragment w którym wyświetlone zostają zdobyte przez użytkownika punkty.

Dodatkowo użytkownik posiada dwa przyciski domyślne które pozwalają na nawigację w każdym etapie gry do ekranu początkowego oraz wyjście z gry.

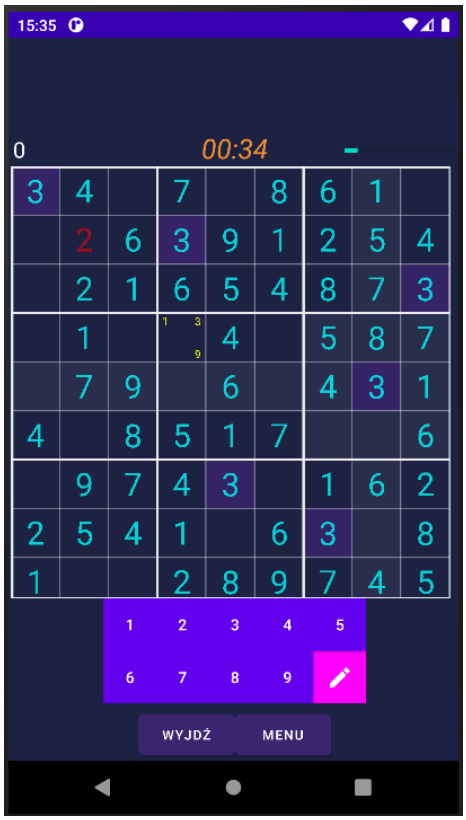
3. Funkcjonalności w aplikacji

Aplikacja posiada wiele elementów posiadających różne funkcjonalności. Warto omówić funkcjonalności bardziej złożonych komponentów.

3.1. SudokuBoardView

SudokuBoardView jest klasą która dziedziczy z View. W ten sposób mamy możliwość przeciążenia podstawowych metod pozwalających na rysowanie własnej planszy oraz obsługiwanie między innymi akcji kliknięcia na planszę. W ten sposób jesteśmy w stanie wykorzystać plansze w interaktywny sposób jednocześnie obsługując dane interfejsu. Klasa implementuje również wykorzystanie Canvas do rysowania w czasie rzeczywistym planszy, a także obsługi interaktywnego interfejsu użytkownika(podświetlanie aktywnych elementów itp.).

Widok SudokuBoardView zaimplementowany w Fragmencie gry wygląda następująco:



Rysunek 1: Widok na planszę sudoku we fragmencie gry

Jak można zaobserwować plansza sudoku posiada podświetlenie oraz różnego rodzaju kolory przy wpisywaniu liczb. Kolor jest zależny od tego czy wartość wpisana jest poprawna bądź czy znajdujemy się w trybie wpisywania notatek.

3.2. Gson

Jest to funkcjonalność pozwalająca na zdekodowanie do programu obiektów zapisanych w pliku o rozszerzeniu .json. Dzięki odczytaniu funkcjonalności jesteśmy w stanie odczytać plansze z zasobów naszej aplikacji i jednocześnie dzięki niej posiadamy możliwość ustalenia tego jak będzie domyślnie wyglądała plansza sudoku.

3.3. GridView i LiveData oraz DataBindings

GridView przechowuje odpowiednio umieszczone przyciski w interfejsie gry aplikacji. Dodatkowo cała aplikacja wykorzystuje odpowiednio ViewModel aby aktualizować UI za pomocą LiveData. LiveData wykrywa zmiany zachodzące w

danych znajdujących się w klasie ViewModel fragmentu poprzez co możemy aktualizować w czasie rzeczywistym stan naszej planszy oraz wielu innych elementów graficznych bez jawnej ich zmiany w klasach widoku(View class).

3.4. Customowy TextClock oraz wielowątkowość programu

W aplikacji wykorzystano również wielowątkowość, której zadaniem jest utworzenie osobnego wątku zliczającego upływ czasu. Upływający czas jest liczony od rozpoczęcia gry w FragmentGame i sam TextClock jest odświeżany w czasie rzeczywistym. Funkcjonalność ta prowadzona jest oczywiście na osobnym wątku nie bez powodu. Wiąże się to z tym że zliczanie co sekundę upływu czasu w wątku głównym programu znacząco obciąży program i istnieje duże ryzyko że sam program zacznie działać mniej płynnie.

3.5. Przekazywanie informacji pomiędzy fragmentami oraz MVVC

Aplikacja wykorzystuje safe args do przekazywania odpowiednich danych ViewModeli do kolejnych części programu, gdzie są one odpowiednio obsługiwane. Dodatkowo cała aplikacja zachowuje zgodność z modelem MVVC, co oznacza że posiada ona realizowane przechowywanie danych oraz interfejs graficzny realizowany w osobnych klasach modelu, które razem zwiększają czytelność kodu, stabilność aplikacji(Nie tracimy danych przy obrocie telefonu itp.).