

**Dokumentacja projektu**

Przedmiot: **Programowanie obiektowe**

**Tytuł projektu:**

**Aplikacja mobilna do nauki języka angielskiego**

**Prowadzący:** **Wykonawca:**

mgr inż. Ewa Żesławska Sandra Paszkiewicz

w64171

3IIZ/2020-GP02

Rzeszów 2022

**Spis treści**

[*1. Opis założeń 3*](#_Toc93260790)

[*2. Specyfikacja wymagań 3*](#_Toc93260791)

[*2.1. Wymagania funkcjonalne 3*](#_Toc93260792)

[*2.2. Wymagania niefunkcjonalne 3*](#_Toc93260793)

[*3. Diagram przypadków użycia 4*](#_Toc93260794)

[*4. Harmonogram realizacji projektu 4*](#_Toc93260795)

[*5. Opis techniczny projektu 5*](#_Toc93260796)

[*6. Prezentacja warstwy użytkowej 11*](#_Toc93260797)

[*7. Raporty z testów jednostkowych 15*](#_Toc93260798)

[*8. System kontroli wersji 16*](#_Toc93260799)

[*9. Dokumentacja projektu 16*](#_Toc93260800)

[*10. Literatura 16*](#_Toc93260801)

# Opis założeń

W dzisiejszych czasach znajomość języka angielskiego jest bardzo ważna, a często osoby młodsze jak i starsze mają problem z jego nauką. Dlatego, aby połączyć przyjemne z pożytecznym powstała ta aplikacja. Przeglądanie słowników niekiedy może stawać się nudne, a jak wiemy telefon posiada teraz prawie każdy, więc w łatwy sposób, by zachęcić się do nauki języka można skorzystać z takiej oto mini gierki. Użytkownik może wybrać sobie tryb, w którym łatwiej jest mu się uczyć. Do wyboru ma dwie opcje: z języka polskiego na angielski lub z języka angielskiego na polski. Po wyborze trybu losowane jest dla niego pojedynczo 10 słówek, a jego zadaniem jest wpisanie tłumaczenia. Jeśli tłumaczenie będzie poprawne wyświetli mu się okno z napisem „Correct!”, w innym przypadku będzie to napis „Wrong!”. Użytkownik ma również możliwość pominięcia słówka, bez udzielenia odpowiedzi. Na sam koniec gracz otrzymuje wynik poprawnych odpowiedzi oraz krótką wiadomość w zależności od uzyskanego wyniku i jeśli ma taką chęć może zacząć grę od nowa lub przejść do ekranu początkowego aplikacji i zmienić tryb nauki na inny. Aplikacja powstała dla urządzeń mobilnych, z racji tego, że telefon nosimy zawsze przy sobie i w każdej chwili, nawet siedząc w autobusie możemy sięgnąć po niego i pouczyć się trochę angielskiego.

# Specyfikacja wymagań

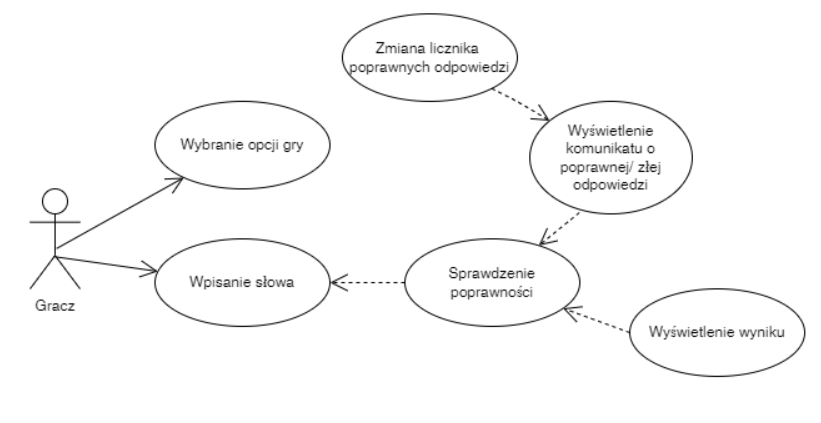
## Wymagania funkcjonalne

* Użytkownik może wybrać tryb nauki języka.
* Losowe słowo jest wyświetlane na ekranie, a gracz może wpisać tłumaczenie w odpowiednim miejscu.
* Użytkownik może pomijać słowa za pomocą przycisku „Submit”.
* Gracz za każde tłumaczenie dostaje informacje czy było ono poprawne.
* Na samym końcu wyświetlany jest wynik oraz krótka wiadomość dla użytkownika, w zależności od uzyskanych punktów.

## Wymagania niefunkcjonalne

* Aplikacja jest przyjazna dla użytkownika i łatwa w użyciu.
* Aplikacja mobilna na systemy Android.
* Gra tworzona w języku Java w środowisku Android Studio.
* Aplikacja nie potrzebuje dostępu do internetu.

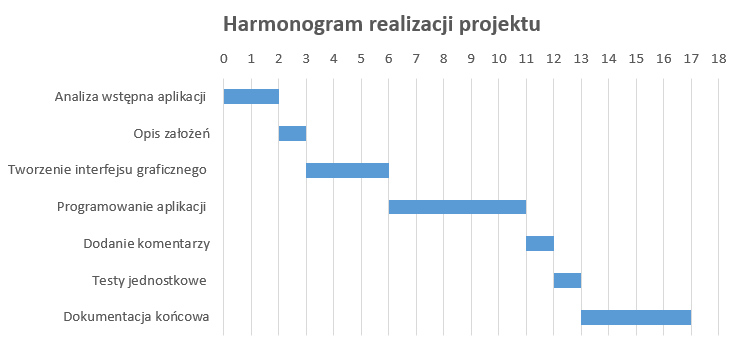
# Diagram przypadków użycia



*Rys. 1. Diagram UML*

# Harmonogram realizacji projektu

Poniżej przedstawiam harmonogram realizacji projektu w postaci diagramu Gantta.



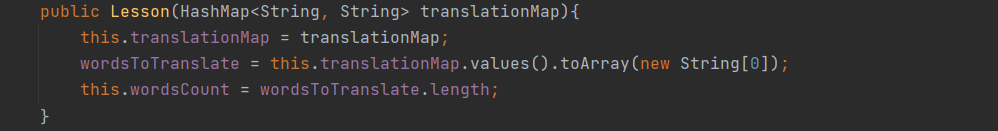
*Rys. 2. Diagram Gantta*

# Opis techniczny projektu

Aplikacja została napisana w języku Java, który jest językiem obiektowym. Kod źródłowy został napisany z podziałem na kilka klas: Lesson, ApplicationState, MainActivity, TranslationMapRandomizer, QuestionActivity, CorrectAnswerActivity, WrongAnswerActivity oraz LessonSummaryActivity.

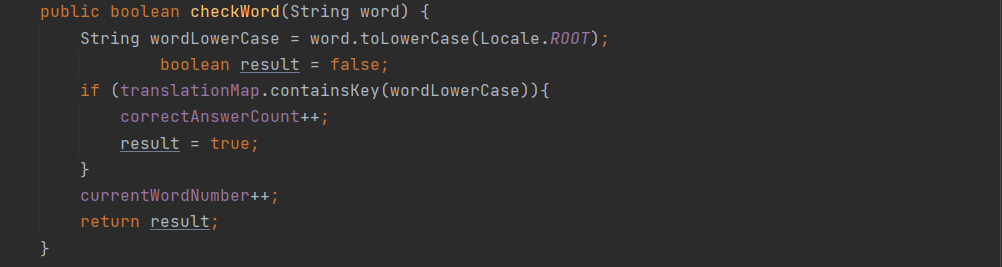
* Środowisko programistyczne: Android Studio Arctic Fox 2020.3.1
* Środowisko programistyczne Javy: Java Development Kit 17.0.1
* Urządzenie z systemem Android 8.0 „Oreo” API 26

*Klasa Lesson.java* – Główna klasa, która zawiera konstruktor lekcji z parametrem mapy, w której kluczem jest słowo podawane przez użytkownika a wartością jest słowo wyświetlane przez aplikację (Rys. 3.).



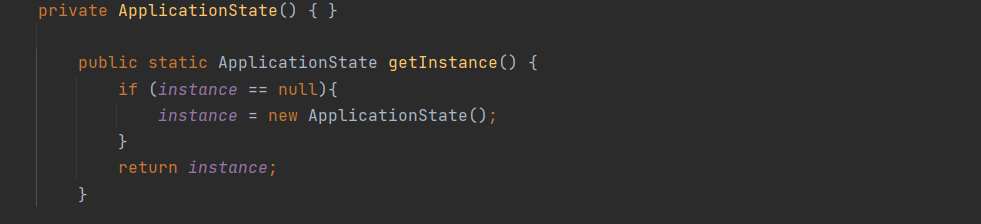
*Rys. 3. Konstruktor lekcji*

Posiada również metodę, która sprawdza, czy użytkownik podał poprawne tłumaczenie i zwraca wartość true jeśli tłumaczenie podane przez użytkownika jest poprawne lub false jeśli nie oraz zwiększa licznik poprawnych odpowiedzi (Rys. 4.). Metoda toLowerCase() konwertuje ciąg na małe litery.



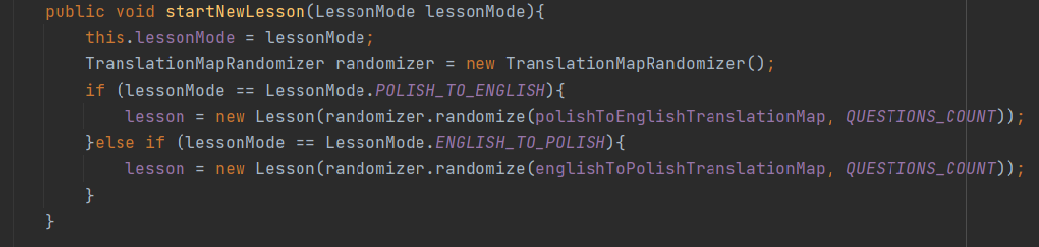
*Rys. 4. Metoda sprawdzająca poprawność tłumaczenia*

*Klasa ApplicationState.java –* Klasa zawiera główny, globalny stan aplikacji służącej do przechowywania aktualnego obiektu lekcji oraz całą mapę tłumaczeń.Każda aktywność ma równy dostęp do tej klasy i danych w niej zawartych. Do stworzenia tej klasy został wykorzystany wzorzec projektowy „Singleton”. Główną cechą klasy stworzonej przy pomocy tego wzorca jest to, że posiada ona tylko jedną instancję na cały program. Efekt ten jest osiągany poprzez stworzenie prywatnego konstruktora (Rys. 5.). Natomiast dostęp do instancji uzyskujemy przy pomocy metody getInstance().



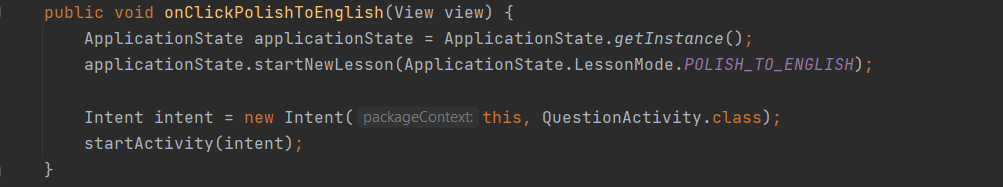
*Rys. 5. Prywatny konstruktor Applicationstate*

Tworzy nowy obiekt lekcji z losową mapą tłumaczeń w danym trybie (Rys. 6.).

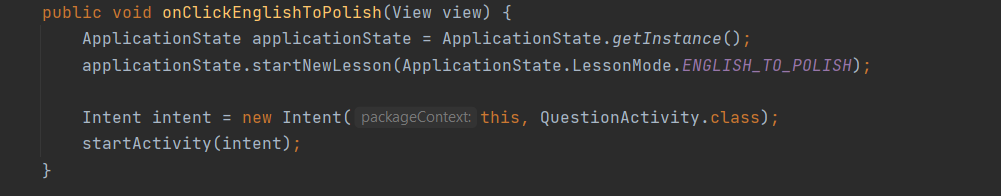


*Rys. 6. Nowy obiekt lekcji*

*Klasa MainActivity.java –* Jest ekranem początkowym aplikacji, który zawiera dwa przyciski, z wyborem trybu lekcji. Pierwszy to tłumaczenie z polskiego na angielski (Rys. 7.), a drugi to tłumaczenie z angielskiego na polski (Rys. 8.). Przy kliknięciu przycisku użytkownik jest przekierowany do QuestionActivity.class.

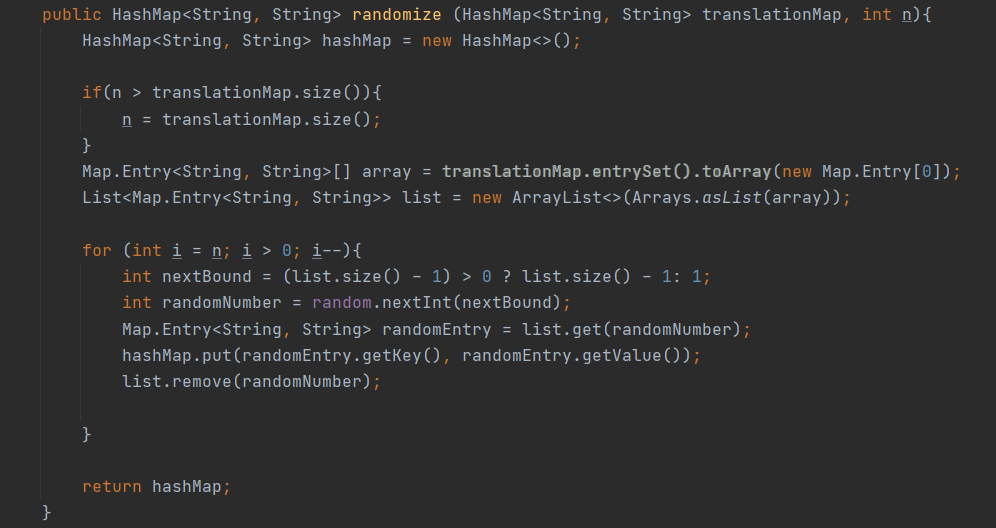


*Rys. 7. Funkcja wywoływana po kliknięciu przycisku „Polish to english”*



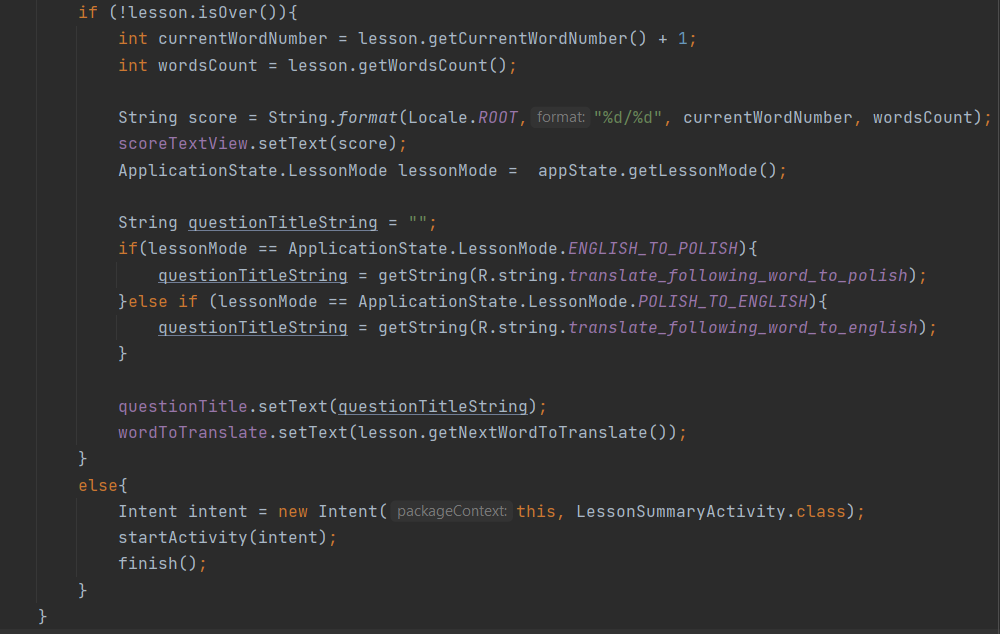
*Rys. 8. Funkcja wywoływana po kliknięciu przycisku „English to polish”*

*Klasa TranslationMapRandomizer.java –* jest to klasa, która posiada metodę wyodrębniania losowych słów dla gracza z mapy tłumaczeń. Sprawdzany jest w niej również parametr n, czyli ilość słów, którą aplikacja ma wylosować. Jeśli jest większy od mapy tłumaczeń to zostaje nadpisany przez translationMap.size(). Zwraca mapę zawierającą n losowych słów z translationMap (Rys. 9.).



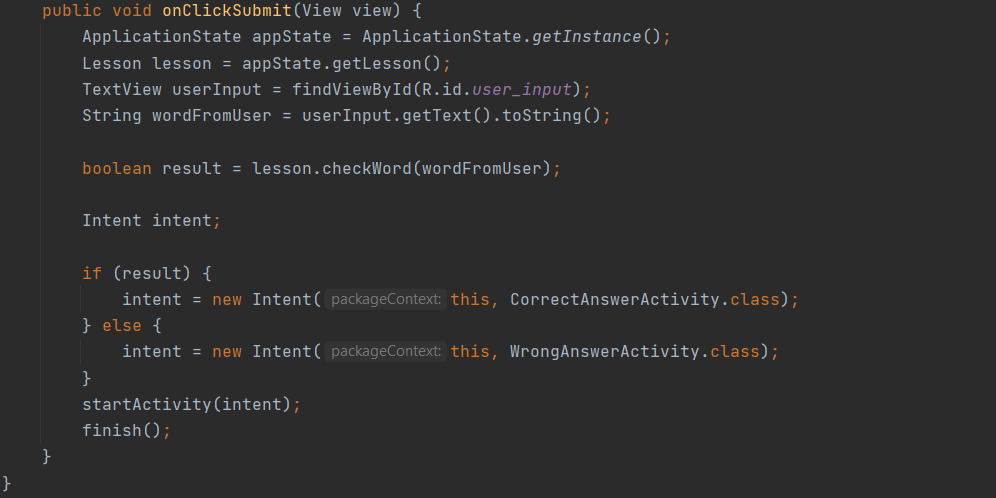
*Rys. 9. Metoda wyodrębniająca losowe słowa dla gracza*

*Klasa QuestionActivity.java –* wyświetla słówko i umożliwia użytkownikowi wpisanie jego tłumaczenia. Słowa są wyświetlane dopóki lekcja nie dobiegnie końca (if (!lesson.isOver())), w przeciwnym przypadku użytkownik zostanie przeniesiony do LessonSummaryActivity. Funkcja wypisuje odpowiedni tekst w zależności od wybranej opcji gry (Rys. 10.).



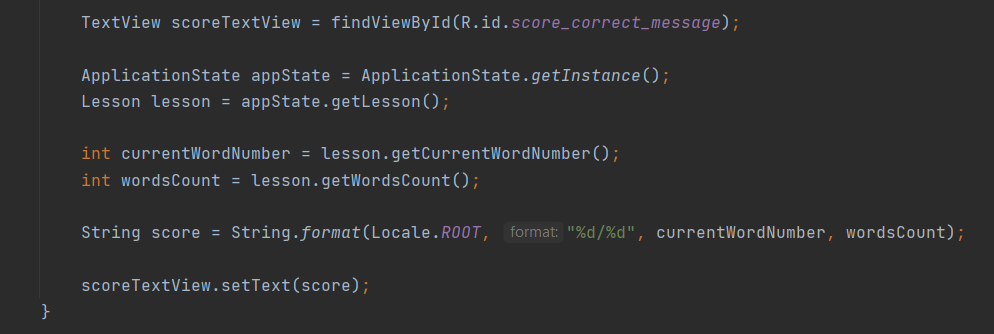
*Rys. 10. Metoda wyświetlająca słowa jeśli lekcja nie jest skończona, a jeśli jest przenosi do innej klasy*

Posiada również przycisk, który po kliknięciu wywołuje funkcję sprawdzającą poprawność tłumaczenia. Jeśli tłumaczenie jest poprawne to przenosi użytkownika do klasy CorrectAnswerActivity, jeśli nie to do WrongAnswerActivity (Rys. 11.).



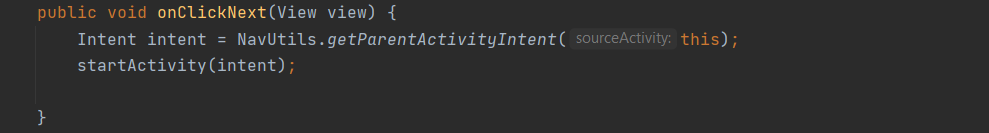
*Rys. 11. Funkcja wywołana po naciśnięciu przycisku „submit”*

*Klasa CorrectAnswerActivity.java -*  aktywność wyświetlana jest, gdy użytkownik udzieli poprawnej odpowiedzi. Zawiera ona wiadomość dla gracza oraz zmienną wyświetlającą aktualny numer pytania i liczbę pytań w prawym górnym rogu (Rys. 12.).



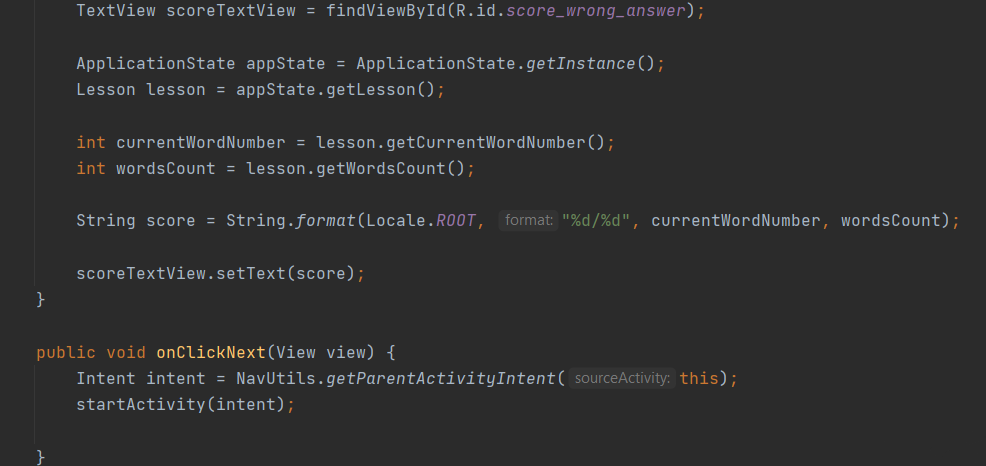
*Rys. 12. Metoda wyświetlająca numer pytania*

Posiada też przycisk „Next”, który przekierowuje użytkownika z powrotem do QuestionActivity. (Rys. 13.).



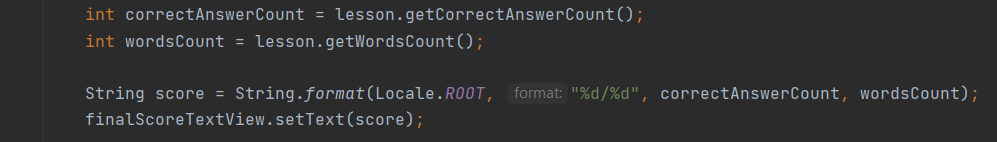
*Rys. 13. Funkcja wywołana, gdy użytkownik naciśnie przycisk „Next”*

*Klasa WrongAnswerActivity.java –* działa dokładnie tak samo jak klasa wyżej, jednakże jest wyświetlana w przypadku, gdy użytkownik udzieli błędnej odpowiedzi (Rys. 14.).



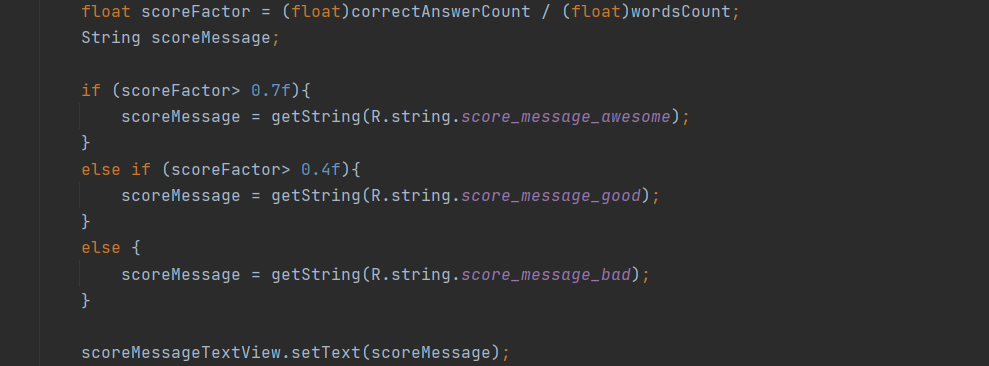
*Rys. 14. Klasa WrongAnswerActivity*

*Klasa LessonSummaryActivity.java –* jest to aktywność wyświetlana na końcu w celu podsumowania całej lekcji. Posiada zmienną wyświetlającą wynik poprawnych odpowiedzi w postaci ułamka (Rys. 15.).



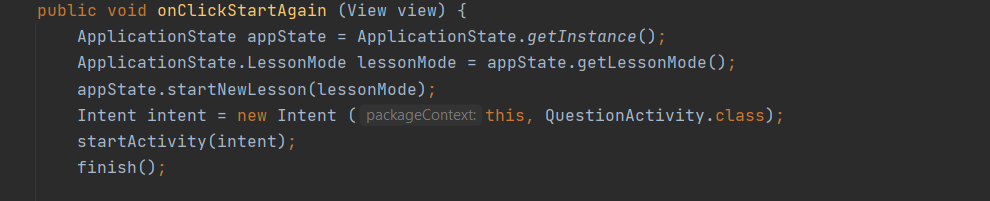
*Rys. 15. Zmienna wyświetlająca wynik*

Występuje w niej również zmienna, która za zadanie ma wypisać wiadomość motywacyjną w zależności od ilości uzyskanych punktów (Rys. 16.).

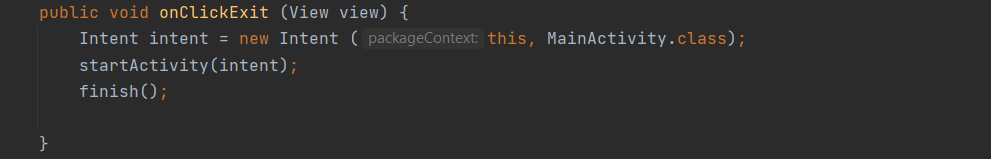


*Rys. 16. Zmienna wypisująca motywacyjną wiadomość w podsumowaniu*

Klasa ta zawiera również dwie funkcje, które są wywoływane po naciśnięciu przycisków. Przycisk „Start Again” rozpoczyna nową lekcję w tym samym trybie, który był wybrany wcześniej i przenosi do klasy QuestionActivity (Rys. 17.). Natomiast przycisk „Exit” przenosi do klasy MainActivity, dzięki czemu użytkownik może zmienić tryb gry (Rys. 18.).



*Rys. 17. Wywołana funkcja po naciśnięciu przycisku „Start Again”*



*Rys. 18. Wywołana funkcja po naciśnięciu przycisku „Exit”*

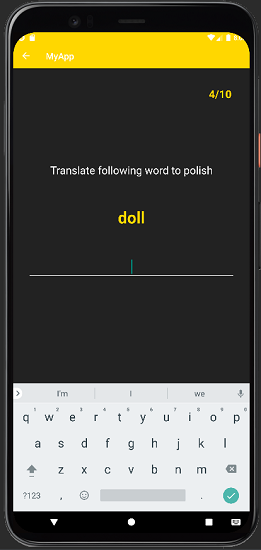
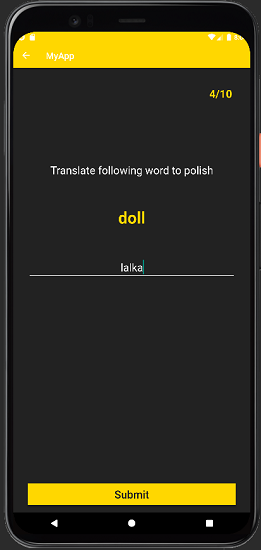
# Prezentacja warstwy użytkowej

Aplikacja stworzona jest tak, by była prosta w obsłudze i bardzo czytelna. Można to od razu zauważyć na ekranie początkowym, który wyświetla się użytkownikowi zaraz po jej otworzeniu (Rys. 19.). Na wyświetlaczu widać napis „Choose lesson” oraz dwa przyciski z dostępnymi opcjami. Gracz może wybrać tryb tłumaczenia wyrazów z języka angielskiego na polski lub z języka polskiego na angielski. Klikając w przycisk przenosi go do gry w wybranej opcji.



*Rys. 19. Ekran początkowy aplikacji.*

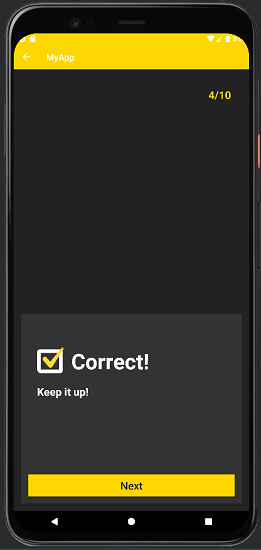
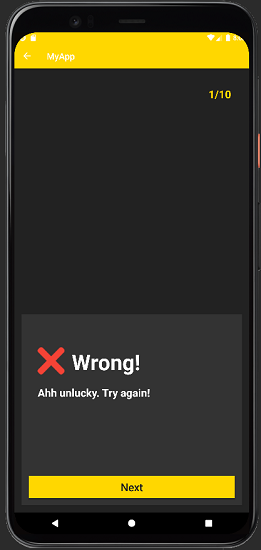
Po wybraniu trybu użytkownikowi pokazuje się okno z pierwszym słówkiem do przetłumaczenia. Słowo, które gracz ma przetłumaczyć wyświetla się w kolorze żółtym, a poniżej znajduje się miejsce na wpisanie swojej odpowiedzi (Rys. 20.). Jeśli użytkownik zatwierdzi słowo przyciskiem na klawiaturze ukaże mu się przycisk „Submit” (Rys. 21.). Klikając w niego program sprawdzi czy słowo zostało poprawnie przetłumaczone. Gracz w jednej rundzie za zadanie ma przetłumaczyć 10 słów, które są losowane przez aplikację. Ilość przetłumaczonych już słów może zobaczyć na liczniku w prawym górnym rogu.

*Rys. 20. Ekran z przykładowym słowem do przetłumaczenia*

*Rys. 21. Ekran wraz z przyciskiem „Submit”*

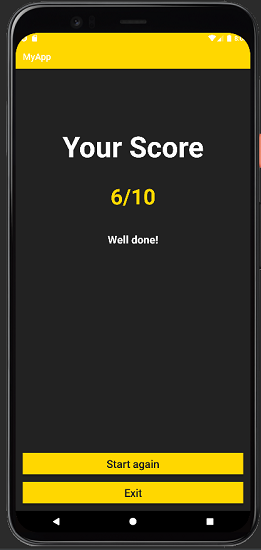
Gdy użytkownik udzieli odpowiedzi i potwierdzi przyciskiem „Submit” aplikacja przeniesie go do innego okna. W zależności od tego czy tłumaczenie było poprawne czy nie pokazuje mu się okno z wiadomością „Correct!” (Rys. 22.) lub „Wrong!” (Rys. 23.). Jeśli naciśnie przycisk „Next” zostanie przeniesiony ponownie do okna ze słowem do przetłumaczenia, jednak licznik słów w prawym górnym rogu podniesie się o 1.

*Rys. 22. Okno po udzieleniu poprawnej odpowiedzi*

*Rys. 23. Okno po udzieleniu błędnej odpowiedzi*

Jeśli licznik słów będzie pokazywał 10/10 to znaczy, że gra dobiegła końca, a użytkownik zostanie przeniesiony do okna z podsumowaniem. Zobaczy tam napis „Your Score” oraz wynik zapisany w postaci ułamka dziesiętnego, który będzie wskazywał na ilość poprawnych odpowiedzi (Rys. 24.). W zależności od ilości uzyskanych punktów, wyświetli się krótka wiadomość dla gracza („Awesome!”, „Well done!”, „Next time will be better!”). Użytkownik na dole ekranu ma do wyboru dwa przyciski: „Start again” – który rozpoczyna grę od nowa; „Exit” – który przenosi gracza do ekranu początkowego, w którym może zmienić tryb gry.



*Rys. 24. Ekran końcowy aplikacji z podsumowaniem wyniku*

# Raporty z testów jednostkowych

W celu sprawdzenia poprawności działania poszczególnych elementów programu przeprowadzone zostały testy jednostkowe w środowisku Android Studio, za pomocą narzędzia JUnit w wersji 4.13.2.

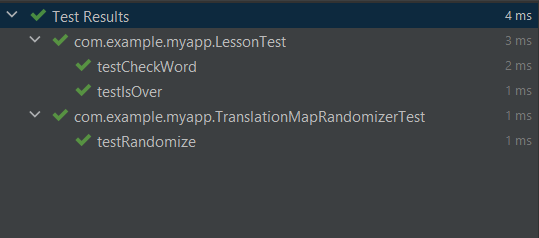
1. Przeprowadzone na klasie Lesson:

* checkWord - jeśli podamy poprawną odpowiedź powinna zwrócić true, a jeśli niepoprawną false.
* isOver – jeśli nie udzielono odpowiedzi na wszystkie pytania powinna zwrócić false, w przeciwnym wypadku true.

1. Przeprowadzone na klasie TranslationMapRandomizer:

* randomize – w przypadku wywołania z argumentem n mniejszym od rozmiaru mapy wejściowej to rozmiar mapy wyjściowej powinien być równy n, a jeśli n jest większy od rozmiaru mapy wejściowej to rozmiar mapy wyjściowej powinien być równy rozmiarowi mapy wejściowej.

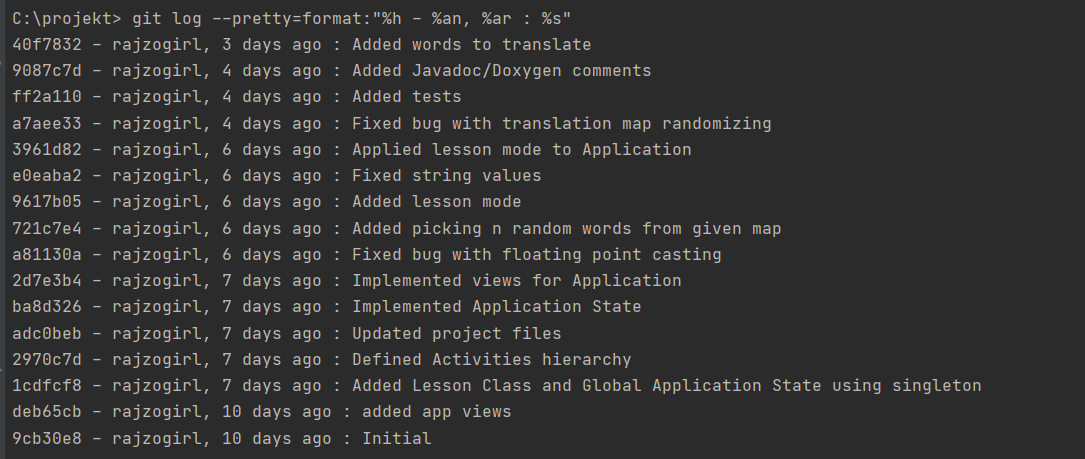
Na rysunku poniżej przedstawione zostały rezultaty wykonanych testów.



*Rys. 25. Rezultaty testów jednostkowych na klasie Lesson oraz TranslationMapRandomizer*

# System kontroli wersji

Projekt został zrealizowany przy pomocy systemu kontroli wersji Git, który służy do zarządzania historią i zabezpiecza codzienną pracę przy kodzie. Na rysunku przedstawiono zrzut ekranowy historii commitów.



*Rys. 26. Historia commitów*

# Dokumentacja projektu

Kod źródłowy został skomentowany po przez system komentarzy dokumentujących Javadoc, a cała dokumentacja projektu wraz z plikami źródłowymi znajduje się w repozytorium na GitHub: <https://github.com/rajzogirl/MyApp>.

# Literatura

1. <https://developer.android.com/guide> (17.01.2022)
2. <https://javastart.pl/baza-wiedzy/android> (17.01.2022)
3. <https://javastart.pl/baza-wiedzy/programowanie-obiektowe> (17.01.2022)
4. <https://www.w3schools.com/java/java_hashmap.asp> (17.01.2022)
5. Git Tutorial for Beginners: <https://youtu.be/DVRQoVRzMIY> (17.01.2022)
6. <https://www.samouczekprogramisty.pl/testy-jednostkowe-z-junit/> (17.01.2022)