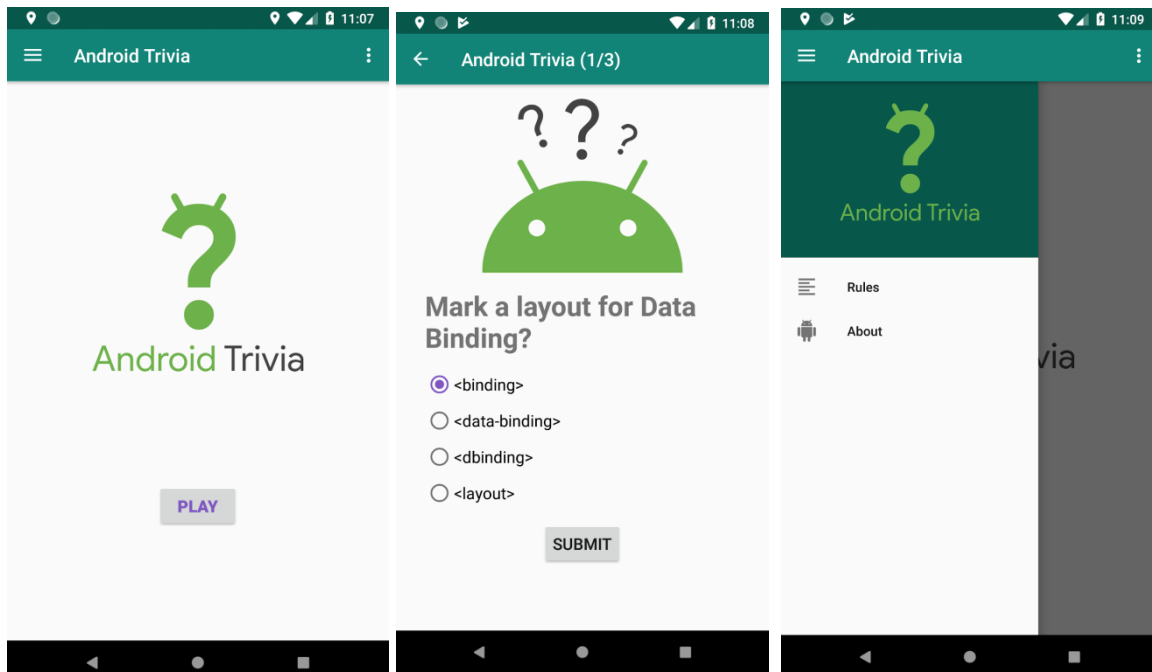



1. Budowa aplikacji w oparciu o fragmenty

2. Opis Aplikacji

W trzech częściach laboratorium stworzymy aplikację o nazwie AndroidTrivia działającą w oparciu o fragmenty. Gotowa aplikacja to gra, w której użytkownik odpowiada na trzy pytania dotyczące kodowania Androida. Jeśli użytkownik odpowie poprawnie na wszystkie trzy pytania, wygrywa grę i może udostępniać swoje wyniki..



Aplikacja AndroidTrivia ilustruje wzorce nawigacji i elementy sterujące. Aplikacja składa się z kilku elementów:

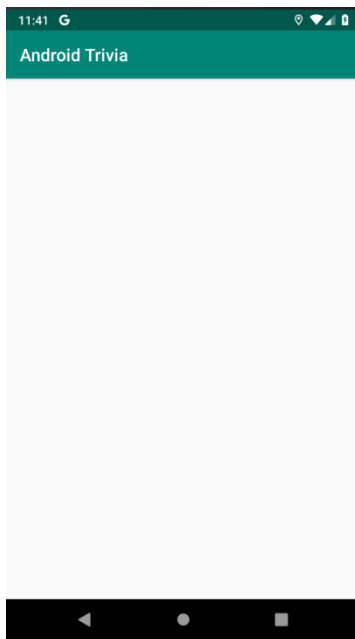
- Na ekranie tytułowym pokazanym po lewej stronie na zrzucie ekranu użytkownik rozpoczyna grę.
- Na ekranie gry z pytaniami pokazanymi na środku powyżej użytkownik gra w grę i przesyła odpowiedzi.
- (navigation drawer) Szuflada nawigacji pokazana po prawej stronie wysuwa się z boku aplikacji i zawiera menu z nagłówkiem. Ikona szuflady  otwiera szufladę nawigacji. Menu szuflady nawigacji zawiera link do strony Informacje i link do reguł gry.

W górnej części aplikacji wyświetlany jest kolorowy widok zwany paskiem aplikacji *app bar*, który jest również znany jako pasek akcji *action bar*.

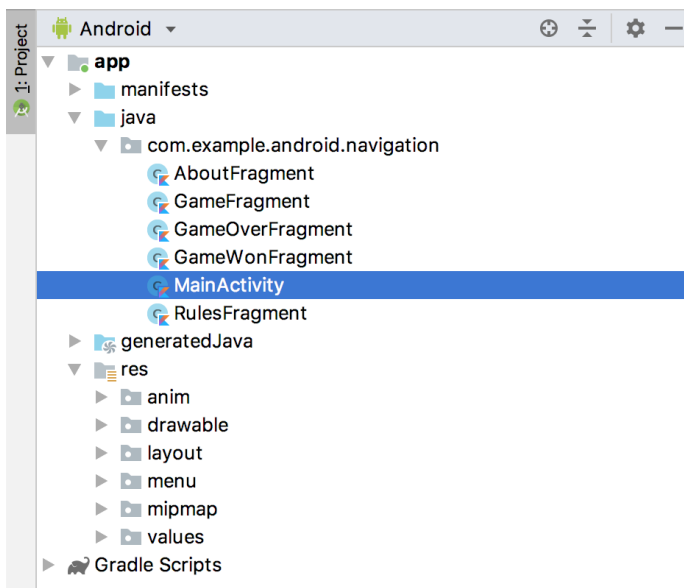
3. Zadanie: poznaj projekt aplikacji startowej

W tym ćwiczeniu korzystamy z aplikacji startowej, która zawiera szablonu kodu i odpowiednie klasy fragmentów.

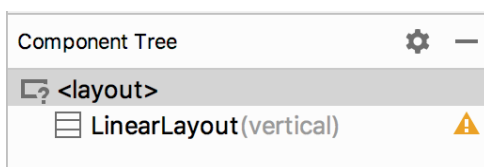
1. Pobierz projekt Startowy.
2. Otwórz projekt w Android Studio i uruchom aplikację. Po otwarciu aplikacja nie robi nic poza wyświetleniem nazwy aplikacji i pustego ekranu.



3. W okienku Projekt Android Studio otwórz widok Projekt: Android, aby eksplorować pliki projektu. Otwórz folder **app** > **java** aby zobaczyć MainActivity class i fragment classes.



4. Otwórz folder **res** > **layout** i kliknij dwukrotnie plik **activity_main.xml**. Plik activity_main.xml pojawia się w Edytorze układu.
5. Kliknij kartę **Design, Component Tree** dla pliku activity_main.xml pokazuje root layout jako pionowy układ liniowy vertical LinearLayout.



W pionowym układzie liniowym wszystkie widoki potomne w układzie są wyrównane w pionie.

4. Zadanie: dodaj fragment

Fragment reprezentuje zachowanie lub część interfejsu użytkownika (UI) w activity. Możesz łączyć wiele fragmentów w jednym activity, aby zbudować wielopanelowy interfejs użytkownika, i możesz ponownie użyć fragmentu w wielu activity.

Pomyśl o fragmencie jako o modułowej części activity, coś w rodzaju „podactivity”, którego możesz również użyć w innych activity:

- Fragment ma własny cykl życia i odbiera własne zdarzenia wejściowe.
- Możesz dodać lub usunąć fragment podczas działania activity.
- Fragment jest zdefiniowany w klasie Kotlin.
- Interfejs użytkownika fragmentu jest zdefiniowany w pliku XML layout.

Aplikacja AndroidTrivia ma główną aktywność i kilka fragmentów. Większość fragmentów i ich plików układu została zdefiniowana wcześniej. W tym zadaniu stworzysz fragment i dodasz go do main activity aplikacji.

Krok 1: Dodaj klasę fragmentu (fragment class)

W tym kroku utworzysz pustą klasę TitleFragment. Zaczynij od utworzenia klasy Kotlin dla nowego fragmentu:

1. W Android Studio kliknij dowolne miejsce w panelu Projekt, aby przywrócić fokus do plików projektu. Na przykład kliknij folder **com.example.android.navigation**.
2. Wybierz **File > New > Fragment > Fragment (Blank)**.
3. Jako nazwę fragmentu użyj **TitleFragment**. Wyczyść wszystkie pola wyboru!: **create Layout XML, include fragment factory methods, include interface callbacks**.
4. Kliknij **Finish**.
5. Otwórz plik fragmentu TitleFragment.kt. Plik zawiera metodę [onCreateView\(\)](#) która jest jedną z metod wywoływanych podczas cyklu życia fragmentu.
6. W metodzie `onCreateView()`, usuń `return TextView(activity).apply { setText(...)` zawierająca `setText`. W `onCreateView()` pozostawia tylko następujący kod:

```
override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?,
                          savedInstanceState: Bundle?): View? {
}
```

Utwórz obiekt wiążący

Fragment nie będzie się teraz kompilował. Aby kompilować fragment, musisz utworzyć obiekt wiążący i nadmuchać widok fragmentu (co jest równoważne z użyciem `setContentView()` activity).

1. W metodzie `onCreateView()` `TitleFragment.kt`, utwórz zmienną wiążącą binding
2. Aby nadmuchać widok fragmentu, wywołaj metodę [DataBindingUtil.inflate\(\)](#) na obiekcie Binding fragmentu, którym jest `FragmentTitleBinding`.

Przekaż cztery parametry do metody:

- inflater, czyli `LayoutInflater` używany do nadmuchiwania binding layout.
 - Zasób układu XML układu do nadmuchiwania. Użyj jednego z układów, który jest już dla Ciebie zdefiniowany, `R.layout.fragment_title`.
 - container pojemnik dla nadrzędnej grupy `ViewGroup`. (en parametr jest opcjonalny.)
 - `false` dla wartości `attachToParent`.
2. Przypisz powiązanie, które `DataBindingUtil.inflate` zwraca do zmiennej powiązania. `binding`.
 3. Zwróć `binding.root`. Teraz kod metody `onCreateView()` wygląda tak:

```
override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?,
    savedInstanceState: Bundle?): View? {
    val binding = DataBindingUtil.inflate<FragmentTitleBinding>(inflater,
        R.layout.fragment_title, container, false)
    return binding.root
}
```

Krok 2: Dodaj nowy fragment do głównego pliku layout

W tym kroku dodajesz `TitleFragment` do pliku `activity_main.xml` layout aplikacji.

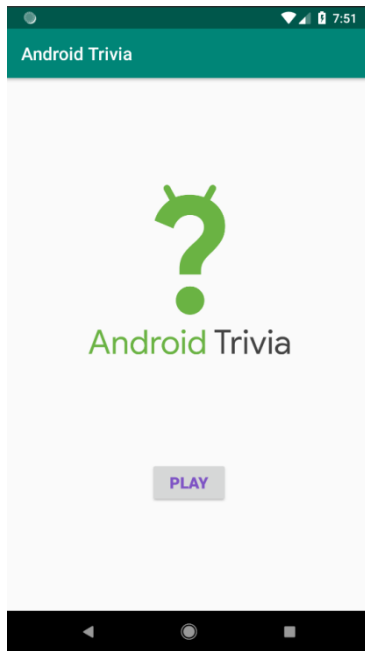
1. Otwórz **res > layout > activity_main.xml** i kliknij kartę **Text** aby wyświetlić kod layout XML.
2. Wewnątrz istniejącego elementu `LinearLayout` dodaj element `fragment`.
3. Ustaw identyfikator fragmentu na `titleFragment`.
4. Ustaw nazwę fragmentu na pełną ścieżkę klasy fragmentu, która w tym przypadku to `com.example.android.navigation.TitleFragment`.
5. Ustaw szerokość i wysokość układu na `match_parent`.

```
<layout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">
        <fragment
            android:id="@+id/titleFragment"
            android:name="com.example.android.navigation.TitleFragment"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
        />
    </LinearLayout>

</layout>
```

6. Uruchom aplikację. Fragment został dodany do ekranu głównego.



W tym ćwiczeniu dodałeś fragment do aplikacji AndroidTrivia, nad którym będziesz dalej pracować w kolejnych dwóch ćwiczeniach.

- *fragment* Fragment jest modułową sekcją activity.
- fragment ma własny cykl życia i odbiera własne zdarzenia wejściowe (input events).
- Użyj znacznika `<fragment>` aby zdefiniować układ fragmentu w pliku układu XML (XML layout).
- Inflate Napompuj układ dla fragmentu w `onCreateView()`.
- Możesz dodać lub usunąć fragment podczas działania activity (while the activity is running).