

# Programowanie urządzeń mobilnych

Podstawowe elementy GUI cz. 2 i 3 – zadania

# Pliki zasobów

- ▶ Jak już robiliśmy wcześniej – sporo rzeczy w Androidzie umieszczamy w plikach xml (np. strings.xml).
- ▶ Jest to dobra praktyka, w plikach zasobów możemy umieszczać sporo rzeczy, do których chcemy mieć łatwy dostęp ze wszystkich miejsc w aplikacji

# Pliki zasobów

- ▶ Elementy:
  - Ciągi znaków
  - Tablice
  - Kolory
  - Wymiary
  - Określenie tła
  - itp.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3   <color name="colorPrimary">#3F51B5</color>
4   <color name="colorPrimaryDark">#303F9F</color>
5   <color name="colorAccent">#FF4081</color>
6 </resources>
```

# Edytor ciągów

Edit translations for all locales in the translations editor.

[Open editor](#) [Hide notification](#)

resources

```
<resources>
  <string name="app_name">Wear.MessageAPI</string>
  <string name="action_settings">Settings</string>
</resources>
```

Show only keys needing translations ?

[Order a translation...](#)

Key	Untranslatable	Default Value
action_settings	<input type="checkbox"/>	Settings
app_name	<input type="checkbox"/>	Wear.MessageAPI

Key:

Default Value:

Resource Folder:

# ImageView

- ▶ Komponent do wyświetlania obrazków na aktywności.
- ▶ Obrazki mogą być przechowywane jako zasoby aplikacji (w katalogu drawable)



```
<ImageView  
    android:id="@+id/imageView1"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_alignParentTop="true"  
    android:layout_centerHorizontal="true"  
    android:layout_marginTop="74dp"  
    android:src="@drawable/ic_launcher" />
```

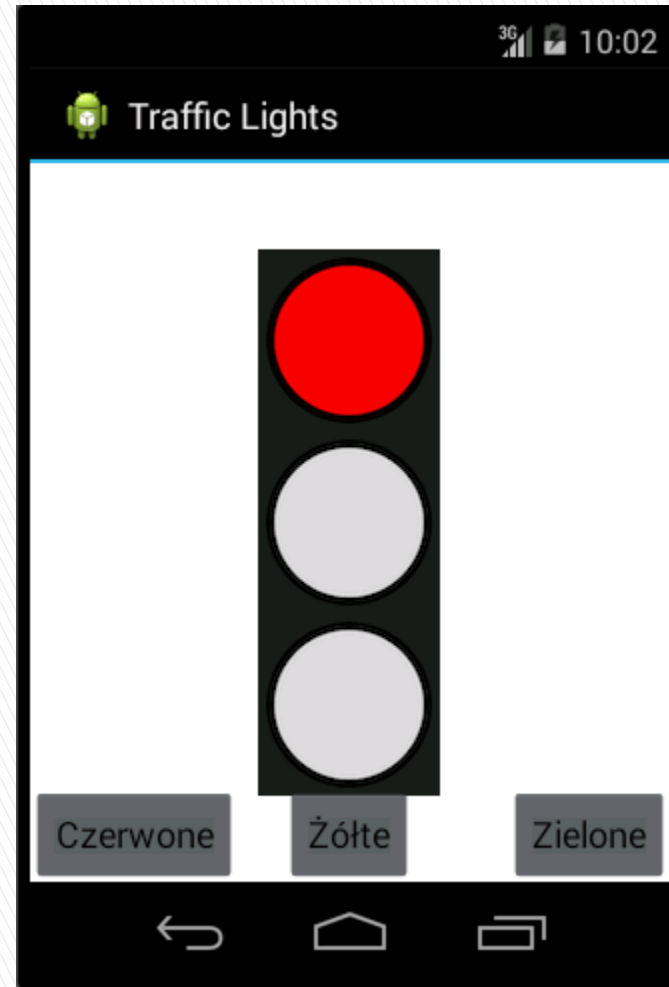
- ▶ By zmienić obrazek można użyć:

```
ImageView im = (ImageView) findViewById(R.id.imageView1);  
im.setImageResource(R.drawable.testimage);
```

# Ćwiczenie 1

- ▶ Napisz aplikację symulującą zachowanie świateł drogowych.
- ▶ Aplikacja powinna wykorzystywać komponenty ImageView i Button.
- ▶ Po kliknięciu na przycisk zapala się odpowiednie światło.
- ▶ Zadbaj o kontrolę błędów – nie można zapalić trzech świateł jednocześnie.
- ▶ Zmiany obrazka dla ImageView dokonuje się funkcją `setImageResource`
- ▶ Obrazki:

[www.tomaszx.pl/materialy/swiatla\\_drogowe\\_obrazki.zip](http://www.tomaszx.pl/materialy/swiatla_drogowe_obrazki.zip)



# Ćwiczenie 2

- ▶ Oprogramuj grę w kółko i krzyżyk dla dwóch osób.
- ▶ W grze ma występować plansza oraz naprzemienne ruchy graczy. Po kliknięciu na dane pole – powinno ono się zmienić na kółko lub krzyżyk odpowiedniego gracza
- ▶ Gra powinna zliczać liczbę wygranych każdego gracza



# Ćwiczenie 3


- ▶ Przygotuj aplikację–quiz na wybrany przez siebie temat.
- ▶ Aplikacja ma wyświetlać zdjęcie wybranego obiektu, a użytkownik ma wpisać co to za przedmiot (lub wybrać z kilku opcji zaproponowanych przez Ciebie).
- ▶ Tematyka dowolna: znane osoby, marki zabawek, loga firm, itp.
- ▶ Przy wpisywaniu tekstu – zadbaj o to, aby małe/wielkie litery nie powodowały braku uznania odpowiedzi.

Julia Dietrich 94

TIME LEFT 9

Miguel Berg 112

This royal crab is found in South America. What is it?



Southern king crab

Marbled rock crab

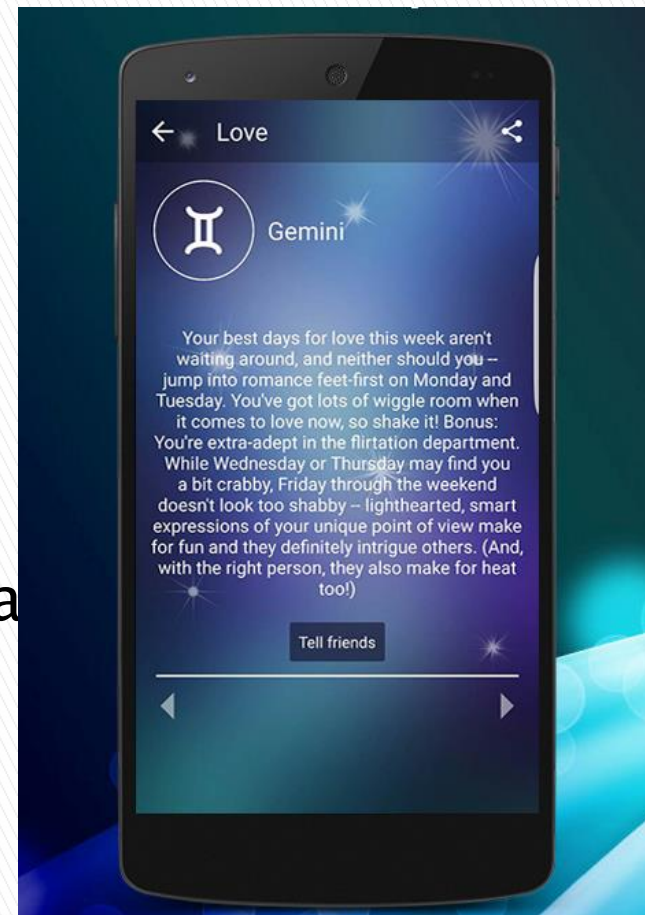
Dungeness crab

Red-clawed crab



# Ćwiczenie 4

- ▶ Wykonaj prostą aplikację typu horoskop.
- ▶ Do pliku strings.xml dodaj tablicę znaków zodiaku.
- ▶ Na formie dodaj element odpytujący użytkownika o datę urodzenia. Za pomocą logiki aplikacji sprawdź jego znak zodiaku.
- ▶ Wyświetl mu horoskop dobrany dla jego znaku zodiaku
- ▶ Dodatkowo, zdefiniuj w pliku zasobów odpowiednie tła dla każdego znaku zodiaku.



# Ćwiczenie 5

- ▶ Na podstawie aplikacji dotyczącej wyboru kawy, zaprojektuj program do zamawiania pizzy.
- ▶ Wymagane funkcjonalności:
  - Wybór typu ciasta (cienkie, grube)
  - Wybór rozmiaru pizzy (mała, średnia, duża)
  - Wybór minimum trzech podstawowych składników z listy dostępnych (szynka, ser, pieczarki, oliwki, boczek, kurczak, cebula)
  - Wybór (opcjonalny) maksymalnie dwóch składników dodatkowych (czosnek, salami, krewetki, kapary, tuńczyk, sos pomidorowy, sos czosnkowy, oregano)
  - Każdy składnik ma swoją cenę, którą należy wliczyć w koszt zamówienia.
  - Aplikacja ma wyświetlać komunikat „Zamówiono małą pizzę na grubym cieście, składającą się z szynki, sera, pieczarek. Cena pizzy to 40 zł.”
  - Przydatne będzie wykorzystanie ScrollView

3G 2:32

GuiDemo

What kind of Coffee?

Decaf

Espresso

Colombian

What else do you like in your coffee?

Cream

Sugar

Pay

Espresso Coffee & Sugar

# Ćwiczenie 6

- ▶ Napisz program, który przekształca skrót państwa (znajdujący się na tablicach rejestracyjnych samochodu) na jego nazwę.
- ▶ Program musi wykorzystywać komponent ListView.
- ▶ Dane odnośnie skrótów i nazw Państw mają być przechowywane w dwóch tablicach np.

```
private String kraje[] =  
{„Polska”, „Niemcy”}  
private String skrot[] = {„PL”,  
„DE”}
```



# Ćwiczenie 6a

- ▶ Zmodyfikuj poprzednie zadanie, by wykorzystywał tablicę ciągów zdefiniowaną w pliku zasobów strings.xml.

- ▶ Odczyt

```
private String countries[];  
countries =  
getResources().getStringArray  
(R.array.countries_array);
```

## Wewnątrz strings.xml

```
<string-array  
name="countries_array">  
    <item>Polska</item>  
    <item>Anglia</item>  
    <item>Niemcy</item>  
    <item>Francja</item>  
    <item>Austria</item>  
    <item>Chorwacja</item>  
    <item>Ukraina</item>  
    <item>Węgry</item>  
</string-array>  
<string-array  
name="shortcuts_array">  
    <item>PL</item>  
    <item>EN</item>  
    <item>DE</item>  
    <item>FR</item>  
    <item>A</item>  
    <item>HR</item>  
    <item>AU</item>  
    <item>H</item>  
</string-array>
```