

## **INF.04: Programowanie aplikacji mobilnych z elementami interfejsu użytkownika**

### **Zadanie egzaminacyjne**

**Temat:** Utwórz aplikację mobilną wyszukującą ukrytą liczbę w układzie siatki.

#### **Opis zadania:**

Twoim zadaniem jest stworzenie aplikacji na system Android, w której użytkownik będzie szukał ukrytej liczby w siatce przycisków. Po naciśnięciu na przycisk, użytkownik odkrywa przypisaną do niego losową liczbę z zakresu od 1 do 100. Gra kończy się, gdy użytkownik znajdzie liczbę większą niż 90.

#### **Wymagania funkcjonalne:**

##### **1. Układ interfejsu użytkownika:**

- Siatka (GridLayout) składająca się z 16 przycisków, ułożonych w układzie 4x4.
- Przyciski oznaczone symbolami „?” przed odkryciem liczby.
- Przycisk „Reset”, umożliwiający ponowne uruchomienie gry z nowymi liczbami.

##### **2. Logika gry:**

- Po naciśnięciu na przycisk w siatce, wyświetla się losowa liczba przypisana do tego przycisku.
- Jeśli odkryta liczba jest większa niż 90, gra się kończy, a użytkownik otrzymuje komunikat: „Gratulacje, znalazłeś ukrytą liczbę!”.
- Gra powinna działać do momentu znalezienia liczby spełniającej warunek lub do ręcznego zresetowania gry.

##### **3. Obsługa przycisku „Reset”:**

- Przywraca stan początkowy gry (ukrywa liczby, przypisuje nowe losowe wartości do przycisków, aktywuje wszystkie przyciski).

#### **Wymagania techniczne:**

##### **1. Struktura kodu:**

- Klasa MainActivity zarządza logiką gry oraz interfejsem użytkownika.
- Metoda setupGame() inicjalizuje grę i przypisuje losowe liczby do przycisków.
- Metoda onClick() obsługuje kliknięcia na przyciski w siatce.

## **2. Losowanie liczb:**

- Do każdego przycisku przypisywana jest losowa liczba z zakresu od 1 do 100.

## **3. Interakcja z użytkownikiem:**

- Komunikat Toast po znalezieniu liczby większej niż 90.

## **Dokumentacja projektu:**

### **1. Plik XML układu interfejsu użytkownika (activity\_main.xml)**

- Powinien zawierać elementy GridLayout dla przycisków oraz przycisk Reset.

### **2. Kod źródłowy aplikacji (MainActivity.java)**

- Zawiera implementację opisanej logiki aplikacji.

## **Kryteria oceny:**

- Poprawność działania aplikacji zgodnie z opisem.
- Czytelność kodu i poprawna organizacja struktury projektu.
- Zgodność z zasadami programowania obiektowego oraz obsługi interfejsu użytkownika w systemie Android.

## **Struktury pliku XML :**

### **1. Układ główny:**

- Używamy Linearnego z orientacją pionową
- Przycisk "Resetuj grę" umieszczamy na górze, a poniżej niego GridLayout z przyciskami.

- Margines wewnętrzny na 16dp

### Siatka (GridLayout):

- Ustawiamy GridLayout z 4 kolumnami i 4 wierszami, aby utworzyć siatkę 4x4.
- Wartość `android:layout_weight="1"` pozwala siatce wypełnić przestrzeń dostępną w LinearLayout, a `android:layout_height="0dp"` zapewnia elastyczność w rozmiarze.

### Styl dla przycisków w siatce:

W pliku `res/values/styles.xml` należy zdefiniować wspólny styl dla przycisków w siatce.

Widok po uruchomieniu



Widok po trzech kliknięciach na znaki zapytania



Widok po trafieniu na liczbę > 90



Widok po resecie



```
<GridLayout
```

16 

```
    android:id="@+id/gridLayout"  
    android:layout_width="match_parent"  
    android:layout_height="0dp"  
    android:layout_weight="1"  
    android:columnCount="4"  
    android:rowCount="4">
```

```
    <!-- Przyciski w siatce -->
```

```
    <Button android:id="@+id/button0" style="@style/GridButton" />  
    <Button android:id="@+id/button1" style="@style/GridButton" />  
    <Button android:id="@+id/button2" style="@style/GridButton" />  
    <Button android:id="@+id/button3" style="@style/GridButton" />  
    <Button android:id="@+id/button4" style="@style/GridButton" />  
    <Button android:id="@+id/button5" style="@style/GridButton" />  
    <Button android:id="@+id/button6" style="@style/GridButton" />  
    <Button android:id="@+id/button7" style="@style/GridButton" />  
    <Button android:id="@+id/button8" style="@style/GridButton" />  
    <Button android:id="@+id/button9" style="@style/GridButton" />  
    <Button android:id="@+id/button10" style="@style/GridButton" />  
    <Button android:id="@+id/button11" style="@style/GridButton" />  
    <Button android:id="@+id/button12" style="@style/GridButton" />  
    <Button android:id="@+id/button13" style="@style/GridButton" />  
    <Button android:id="@+id/button14" style="@style/GridButton" />  
    <Button android:id="@+id/button15" style="@style/GridButton" />
```

```
</GridLayout>
```

```
<resources>
```

```
    <style name="GridButton">
```

```
        <item name="android:layout_width">0dp</item>
```

```
        <item name="android:layout_height">wrap_content</item>
```

```
        <item name="android:layout_columnWeight">1</item>
```

```
        <item name="android:layout_rowWeight">1</item>
```

```
        <item name="android:textSize">24sp</item>
```

```
        <item name="android:textColor">@android:color/black</item>
```

```
        <item name="android:backgroundTint">@android:color/white</item>
```

```
    </style>
```

```
</resources>
```

```
// Inicjalizacja przycisków
buttons = new Button[16];
for (int i = 0; i < 16; i++) {
    String buttonID = "button" + i;
    int resID = getResources().getIdentifier(buttonID, defType: "id", getPackageName())
    buttons[i] = findViewById(resID);
}
```

```
private void setupGame() {
    randomNumbers = new int[16];
    Random random = new Random();

    for (int i = 0; i < 16; i++) {
        randomNumbers[i] = random.nextInt( bound: 100) + 1;
        Button button = buttons[i];
        button.setText("?");
        button.setEnabled(true);
        int finalI = i;
        button.setOnClickListener(v -> onClick(finalI, button));
    }
    gameActive = true;
}
```

```
private void onClick(int index, Button button) {
    if (!gameActive) return;

    // Odkrywamy liczbę
    int number = randomNumbers[index];
    button.setText(String.valueOf(number));
    button.setEnabled(false); // Dezaktywujemy przycisk

    if (number > 90) {
        Toast.makeText( context: this, text: "Gratulacje, znalazłeś ukrytą 1:
        gameActive = false; // Kończymy grę
    }
}
```