Mobile and Wearable Computing

Assignment 02

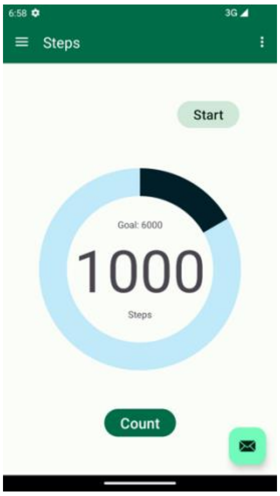
Davide Grandesso

Exercise 1 – Android Basics

1. *What does the "android:minSdkVersion" in android project indicate?*
   - It indicates the minimum version required to run the android project, this is useful because a project uses some features present only by a specific version of android
2. *Why Android documentation indicates that declaring the attribute "android: maxSdkVersion" is not recommended?*
   - "android: maxSdkVersion" is not recommended because it would mean that a user to use an app cannot update the version of android on his device
3. *One of the UI components of android is "Linear layout", what are the types of the "Linear layout"?*
   - **Horizontal (android:orientation="horizontal")** → The elements are arranged in a single row
   - **Vertical (android:orientation="vertical")** → The elements are arranged in a single column
4. *Explain what are the differences between activity and fragment?*
   - A **Fragment** is a reusable part of an **Activity**
     - **Activity →** is a single screen of an application
     - **Fragment**
       - Represents a behavior or a portion of user interface
       - Sub-activity that can be reused in different activities
5. *What are the two types of Navigation Drawer? Explain the differences between the two types?*
   - Navigation drawers contain a list of items, which can be enhanced and organized with headers and dividers, there are two types of Navigation Drawer:
     - **Standard drawer**
       - Allow access drawer destinations and app content
       - Permanently visible or open/close
     - **Modal drawer**
       - Block interaction with the rest of an app's content
       - Primarily used on phones

Exercise 2 – Material Design

1. Add a "count" button

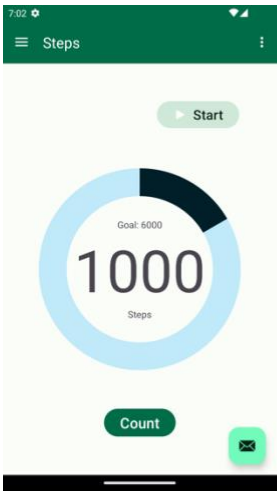


Add a "count" button, like in Tutorial 01. The "count" button should be positioned ***below*** *the Circular Progress Bar. The size and the button type should be appropriate.*

```
<Button
    android:id="@+id/button_count"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="50dp"
    android:text="Count"
    android:textSize="22sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@+id/progressBar" />
```

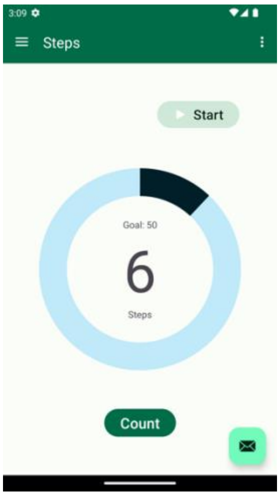
2. Add an icon to the "start" button.



Add an icon to the "start" button. The icon should be appropriate for the intended action.

```
<Button
    android:id="@+id/start_button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="56dp"
    android:backgroundTint="?attr/colorSecondaryContainer"
    android:text="@string/start_text"
    android:textColor="?attr/colorOnSecondaryContainer"
    android:textSize="20sp"
    app:icon="@android:drawable/ic_media_play"
    app:iconSize="22dp"
    app:layout_constraintBottom_toTopOf="@+id/progressBar"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

3. Increase the counter



Make it so that when you press "count" , the counter in the middle increases by one. Here, also modify the app to have the ***Circular Progress Indicator*** *increase with the count number.*

```
private int counter;
private int goal;

counter = 0;
goal = 50;

stepCountsView = (TextView) root.findViewById(R.id.counter);
stepCountsView.setText(Integer.toString(counter));
```

```
Button count = (Button) root.findViewById(R.id.button_count);
count.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        counter++;
        stepCountsView.setText(Integer.toString(counter));
        progressBar.setProgress(counter);
    }
});

progressBar = (CircularProgressIndicator)
root.findViewById(R.id.progressBar);
progressBar.setMax(goal);
progressBar.setProgress(counter);
```

4. Reset to 0

Make it so that when you press the "start" button, the counter is reset to 0.

```
private int counter;


counter = 0;


stepCountsView = (TextView) root.findViewById(R.id.counter);
stepCountsView.setText(Integer.toString(counter));


Button start = (Button) root.findViewById(R.id.start_button);
start.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        counter = 0;
        stepCountsView.setText(Integer.toString(counter));
        progressBar.setProgress(counter);
    }
});
```

5. Change the icon of the app

Change the icon of the app. Choose/create a representative icon for your StepApp. Set the default Android icon to the icon you chose/created.

```
android:icon="@mipmap/stepapp"

android:roundIcon="@mipmap/stepapp_round"
```

Counter restart

```
Button count = (Button) root.findViewById(R.id.button_count);
count.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (counter == goal){
            counter = 0;
        }
        counter++;
        stepCountsView.setText(Integer.toString(counter));
        progressBar.setProgress(counter);


    }
});
```

Toast message

```
Button count = (Button) root.findViewById(R.id.button_count);
count.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (counter == goal){
            counter = 0;
        }
        counter++;
        stepCountsView.setText(Integer.toString(counter));
        progressBar.setProgress(counter);
        if (counter == goal)
            Toast.makeText(getContext(), "Congratulations",
Toast.LENGTH_SHORT).show();

    }
});
```

Repository:

https://github.com/dadegrande99/AssignmentsMWC_Grandesso/tree/master/Assignment02