Bài 1.2 Phần A: Giao diện người dùng tương tác đầu tiên

Giới thiệu

Giao diện người dùng (UI) xuất hiện trên màn hình của một thiết bị Android bao gồm một hệ thống phân cấp các đối tượng gọi là views—mỗi phần tử trên màn hình đều là một View. Lớp View là khối xây dựng cơ bản cho tất cả các thành phần UI và là lớp cơ sở cho các thành phần UI tương tác như nút bấm , kiểm tra, và ô nhập văn bản v.v. Một số lớp con phổ biến của View được sử dụng trong nhiều bài học bao gồm:

- **TextView**: Hiển thị văn bản.
- **EditText**: Cho phép người dùng nhập và chỉnh sửa văn bản.
- **Button** và các phần tử có thể nhấp khác (**RadioButton, CheckBox, Spinner**) để tạo hành vi tương tác.
- **ScrollView** và **RecyclerView**: Hiển thị danh sách có thể cuộn.
- **ImageView**: Hiển thị hình ảnh.
- **ConstraintLayout và LinearLayout**: Chứa các phần tử View khác và sắp xếp vị trí của chúng.

Mã Java để hiển thị và điều khiển UI được chứa trong một lớp mở rộng Activity. Activity thường được liên kết với một tệp XML định nghĩa bố cục của các View trên màn hình.

Ví dụ, , mã MainActivity trong ứng dụng Hello World hiển thị một bố cục được định nghĩa trong tệp activity_main.xml, trong đó có một TextView chứa nội dung "Hello World".

Trong các ứng dụng phức tạp hơn, một Activity có thể thực hiện các thao tác như xử lý các hành động khi người dùng chạm vào màn hình, vẽ nội dung đồ họa, truy vấn dữ liệu từ cơ sở dữ liệu hoặc internet. Bạn sẽ tìm hiểu về lớp Activity chi tiết hơn trong các bài học tiếp theo.

Trong bài thực hành này, bạn sẽ học cách tạo ứng dụng tương tác đầu tiên ứng dụng này cho phép người dùng tương tác với các thành phần trên màn hình. Bạn sẽ tạo một ứng dụng bằng mẫu Empty Activity, học cách sử dụng trình chỉnh sửa bố cục (Layout Editor) để thiết kế giao diện và chỉnh sửa bố cục trong XML. Bjan cần phát triển những kĩ năng vì bạn có thể hoàn thành các bài thực hành khác trong khoá học.

Những gì bạn nên biết trước

Bạn nên có kiến thức về:

- Cách cài đặt và mở Android Studio.
- Cách tạo ứng dụng HelloWorld.
- Cách chạy ứng dụng HelloWorld.

Những gì bạn sẽ học

- Cách tạo một ứng dụng với hành vi tương tác.
- Cách sử dụng trình chỉnh sửa bố cục (Layout Editor) để thiết kế UI.
- Cách chỉnh sửa bố cục trong XML.
- Nhiều thuật ngữ mới về lập trình Android (xem mục "Thuật ngữ và khái niệm" để hiểu rõ hơn).

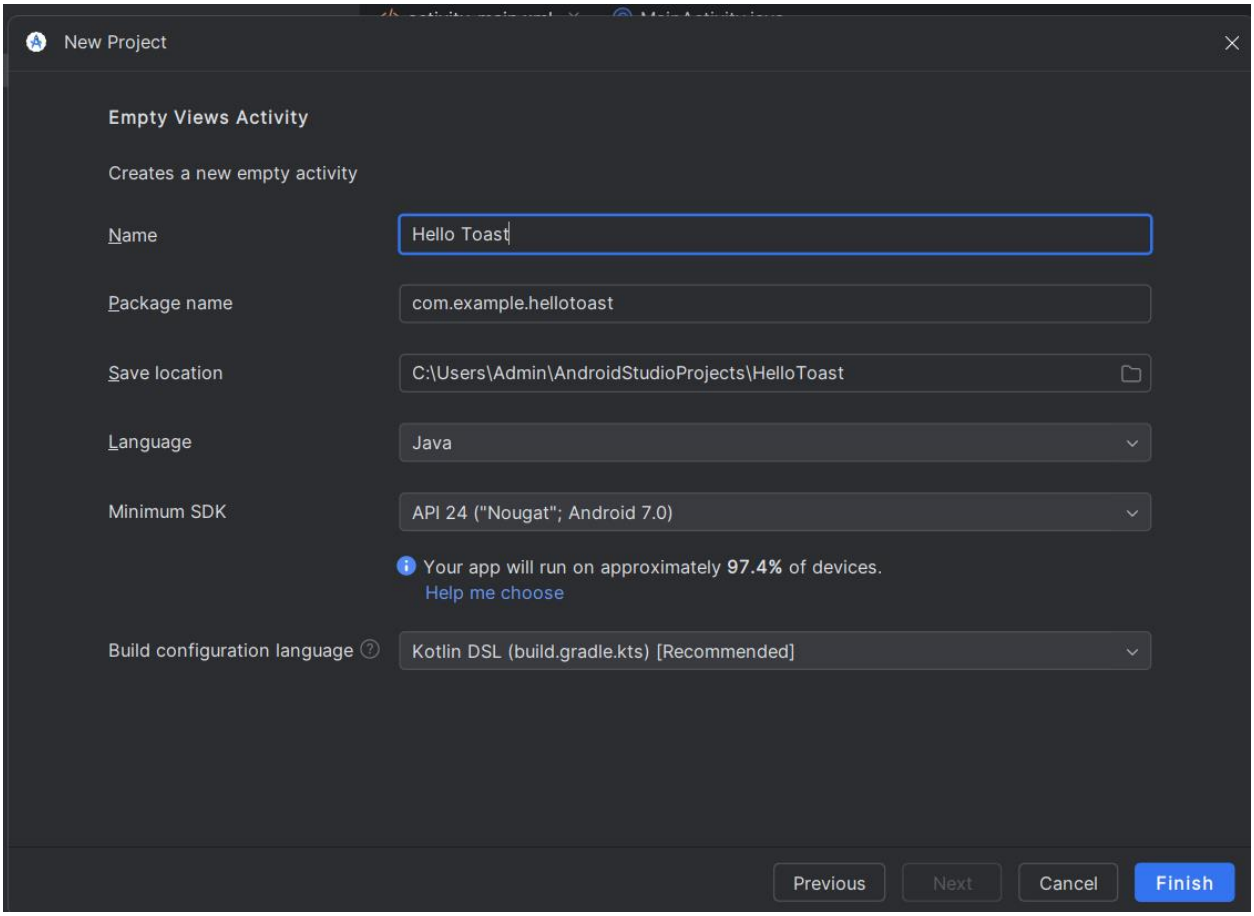
Những gì bạn sẽ làm

- Tạo một ứng dụng và thêm hai nút (Button) cùng một TextView vào bố cục.
- Sắp xếp các phần tử trong ConstraintLayout để căn chỉnh chúng theo lề và các phần tử khác.
- Thay đổi thuộc tính của các phần tử UI.
- Chỉnh sửa bố cục ứng dụng bằng XML.
- Trích xuất các chuỗi văn bản cứng (hardcoded strings) thành tài nguyên chuỗi (string resources).
- Thêm phương thức xử lý sự kiện (click-handler methods) để hiển thị thông báo trên màn hình khi người dùng nhấn vào từng nút.

Nhiệm vụ 1: Tạo và khám phá một dự án mới

Trong bài thực hành này, bạn sẽ thiết kế và triển khai một dự án cho ứng dụng HelloToast**.**
(Một liên kết đến mã nguồn giải pháp sẽ được cung cấp ở cuối bài.)

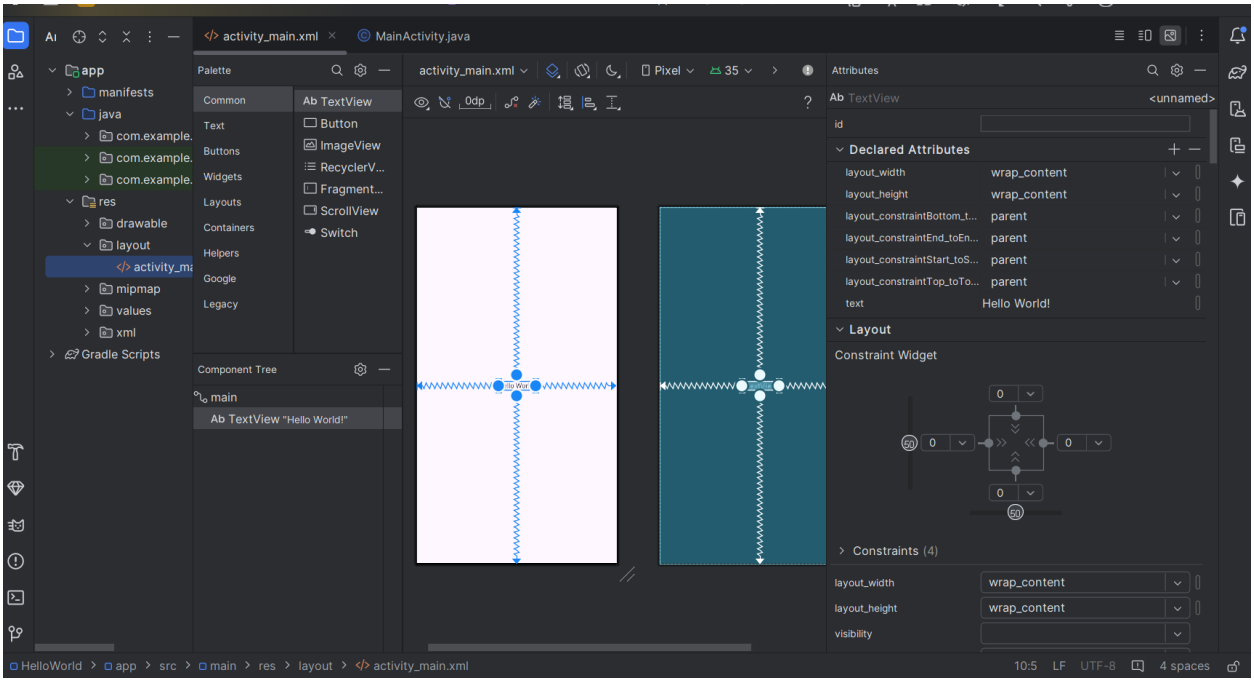
1.1.Tạo dự án trên Android Studio



1.2. Khám phá chỉnh sửa bố bục

Android Studio cung cấp trình soạn thảo bố cục để nhanh chóng xây dựng bố cục giao diện người dùng (UI) của ứng dụng. Nó cho phép bạn kéo các phần tử vào chế độ xem thiết kế trực quan và bản thiết kế, định vị chúng trong bố cục, thêm ràng buộc và đặt thuộc tính. Ràng buộc xác định vị trí của một phần tử UI trong bố cục. Ràng buộc đại diện cho kết nối hoặc căn chỉnh với một chế độ xem khác, bố cục cha hoặc đường hướng dẫn ẩn.

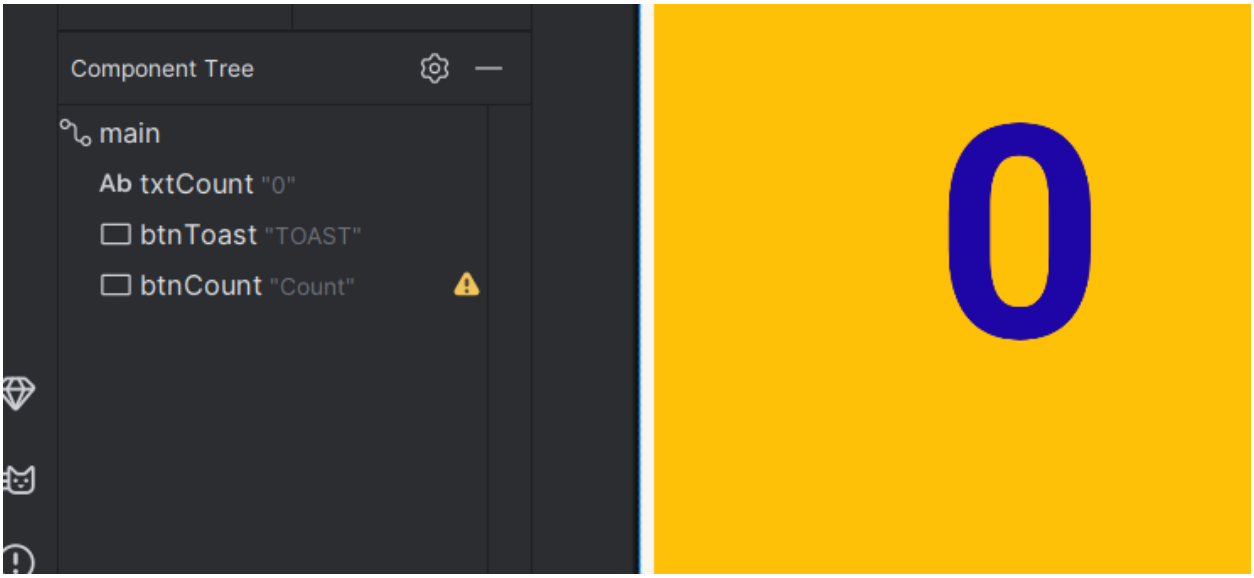
Hãy khám phá trình soạn thảo bố cục và tham khảo hình ảnh bên dưới khi bạn làm theo các bước được đánh số:



1. Trong thư mục app > res > layout trong ngăn Project > Android, nhấp đúp vào tệp activity_main.xml để mở nó, nếu nó chưa được mở.
2. Nhấp vào tab **Design** nếu nó chưa được chọn. Bạn sử dụng tab **Design** để thao tác các phần tử và bố cục, và tab **Text** để chỉnh sửa mã XML cho bố cục.
3. Ngăn **Palette** hiển thị các phần tử giao diện người dùng (UI) mà bạn có thể sử dụng trong bố cục ứng dụng của mình.
4. Ngăn **Component Tree** hiển thị cấu trúc phân cấp chế độ xem của các phần tử UI. Các phần tử chế độ xem được tổ chức thành một cấu trúc phân cấp cây của cha và con, trong đó một con kế thừa các thuộc tính của cha. Trong hình trên, TextView là con của ConstraintLayout. Bạn sẽ tìm hiểu về các phần tử này sau trong bài học này.
5. Ngăn **Design** và **Blueprint** của trình soạn thảo bố cục hiển thị các phần tử UI trong bố cục. Trong hình trên, bố cục chỉ hiển thị một phần tử: một TextView hiển thị "Hello World".
6. Tab **Attributes** hiển thị ngăn **Attributes** để đặt các thuộc tính cho một phần tử UI.

Nhiệm vụ 5: Chỉnh sửa bố cục trong XML

Bố cục ứng dụng Hello Toast gần hoàn thiện! Tuy nhiên, một dấu chấm than xuất hiện bên cạnh mỗi phần tử UI trong Cây thành phần. Di con trỏ qua các dấu chấm than này để xem các thông báo cảnh báo, như hiển thị bên dưới. Cảnh báo tương tự xuất hiện cho cả ba phần tử: chuỗi được mã hóa cứng phải sử dụng tài nguyên.



Cách dễ nhất để khắc phục sự cố bố cục là chỉnh sửa bố cục trong XML. Mặc dù trình chỉnh sửa bố cục là một công cụ mạnh mẽ, nhưng một số thay đổi dễ thực hiện trực tiếp trong mã nguồn XML.

Nhiệm vụ 6: Thêm xử lý onClick cho các nút

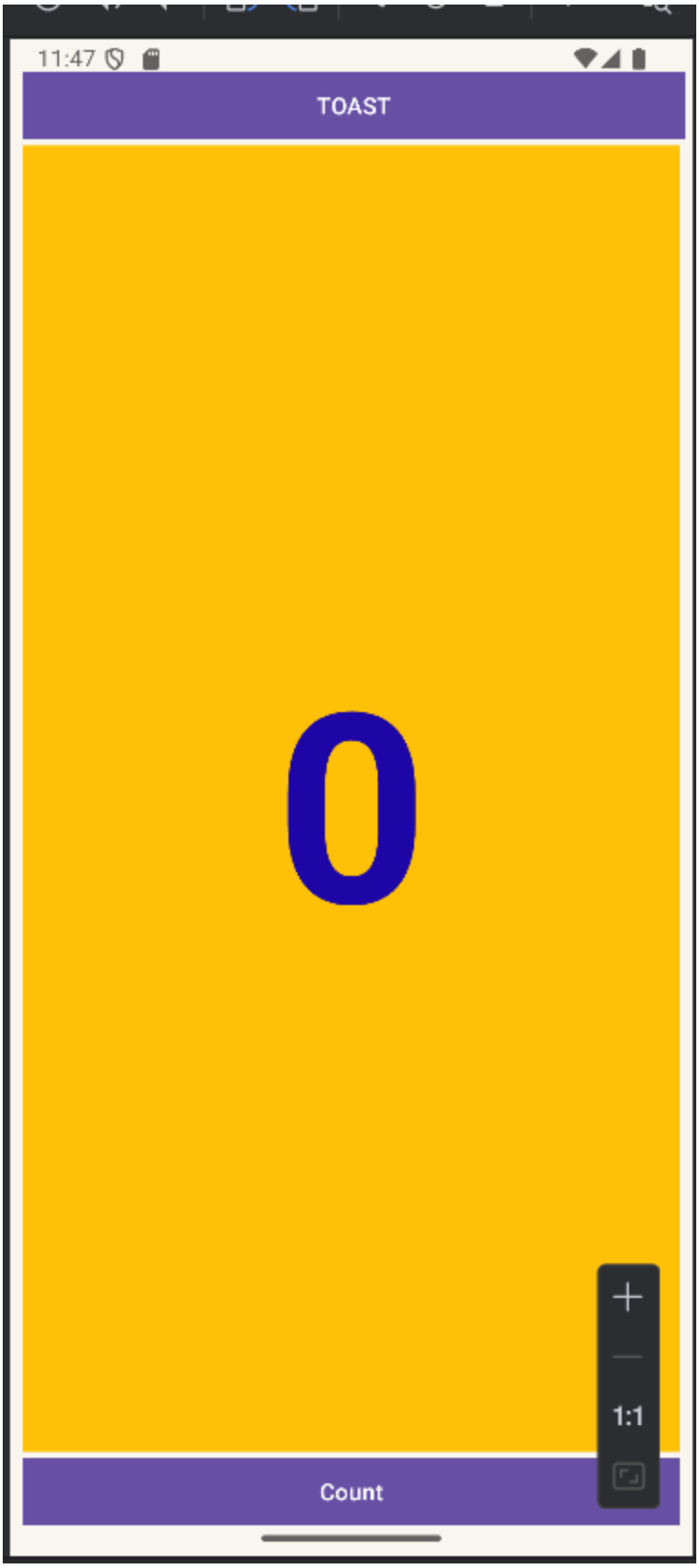
```
    android:layout_marginTop="16dp"
    android:onClick="showToast"
    android:text="TOAST"
```

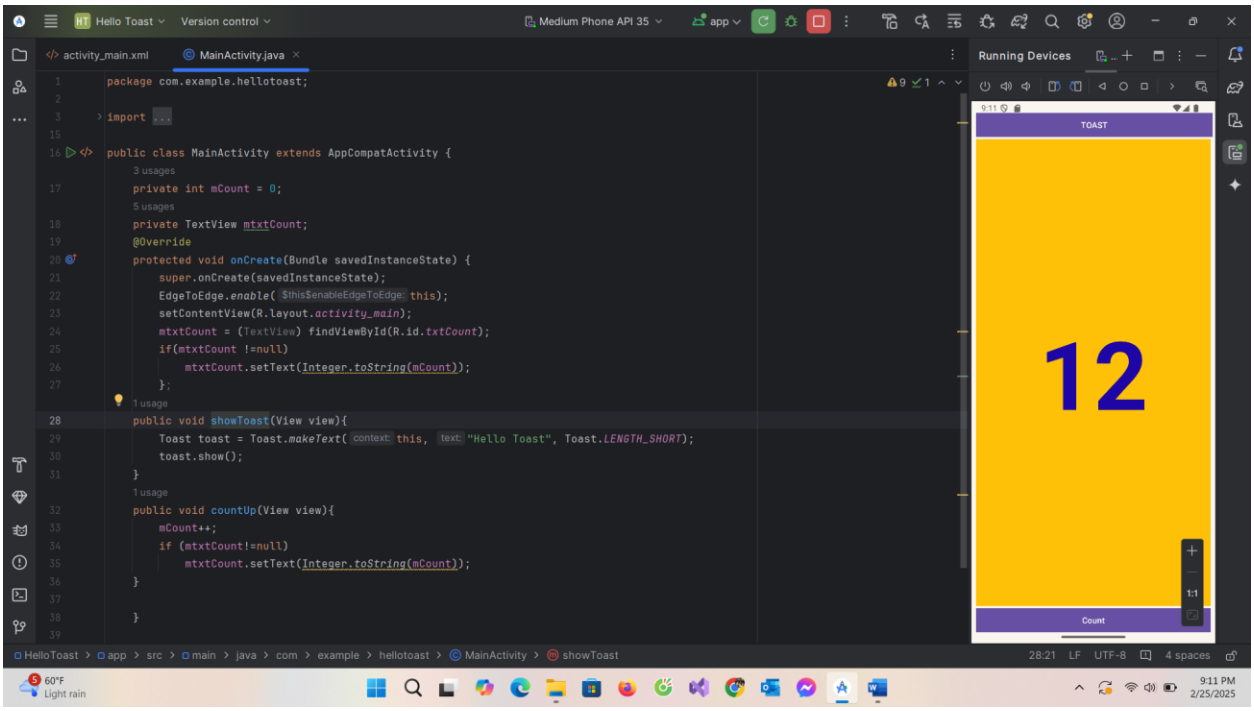
```
    android:layout_marginBottom="16dp"
    android:onClick="countUp"
    android:text="Count"
    app:cornerRadius="0dp"
    app:layout_constraintBottom_toBottomOf="par
```

6.2. Chỉnh sửa trình xử lý Nút Toast

Bây giờ bạn sẽ chỉnh sửa phương thức showToast()—trình xử lý nhấp chuột Nút Toast trong MainActivity—để nó hiển thị một thông báo. Toast cung cấp một cách để hiển thị một thông báo đơn giản trong một cửa sổ bật lên nhỏ. Nó chỉ lấp đầy lượng không gian cần thiết cho thông báo. Hoạt động hiện tại vẫn hiển thị và tương tác. Toast có thể hữu ích để kiểm tra tính tương tác trong ứng dụng của bạn—thêm thông báo Toast để hiển thị kết quả của việc chạm vào Nút hoặc thực hiện một hành động.

Thực hiện theo các bước sau để chỉnh sửa trình xử lý nhấp chuột Nút Toast:
1. Xác định vị trí phương thức showToast() mới tạo.
2. Để tạo một phiên bản của Toast, hãy gọi phương thức makeText() trên lớp Toast
3. Cung cấp ngữ cảnh của Activity ứng dụng. Vì Toast hiển thị ở đầu UI Activity nên hệ thống cần thông tin về Activity hiện tại. Khi bạn đã ở trong ngữ cảnh của Activity mà ngữ cảnh bạn cần, hãy sử dụng điều này như một phím tắt.
4. Cung cấp thông báo để hiển thị, chẳng hạn như tài nguyên chuỗi (taste_message bạn đã tạo ở bước trước). Tài nguyên chuỗi toast_message được xác định bởi R.string.
5. Cung cấp thời lượng hiển thị. Ví dụ: Toast.LENGTH_SHORT hiển thị toast trong thời gian tương đối ngắn.
Thời lượng hiển thị Toast có thể là Toast.LENGTH_LONG hoặc Toast.LENGTH_SHORT. Độ dài thực tế là khoảng 3,5 giây cho Toast dài và 2 giây cho Toast ngắn.
6. Hiển thị Toast bằng cách gọi show(). Sau đây là toàn bộ phương thức showToast():
Chạy ứng dụng và xác minh rằng thông báo Toast xuất hiện khi nhấn nút Toast.

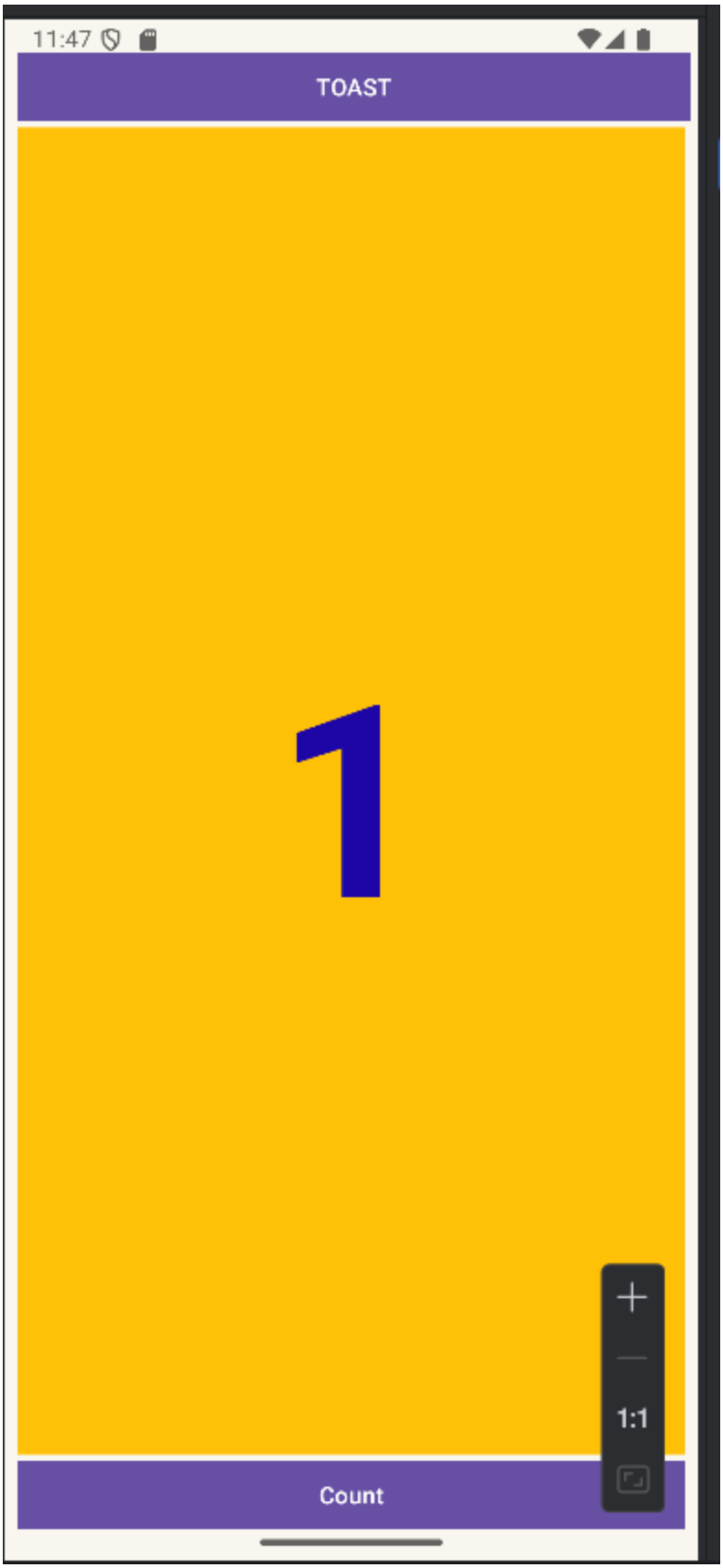

11:47
TOAST
0
1:1
Count

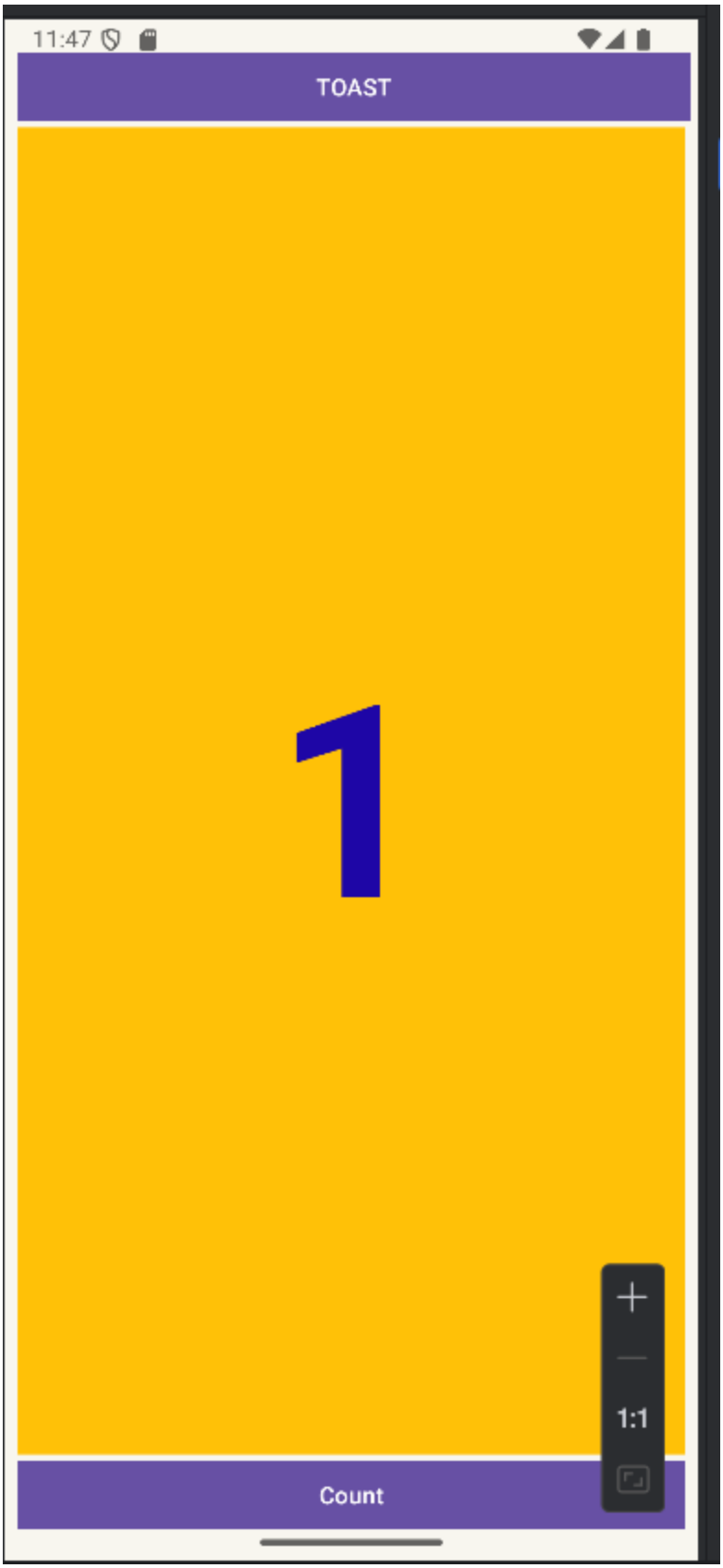


6.3. Chỉnh sửa trình xử lý Nút Count

```java
public class MainActivity extends AppCompatActivity {
    3 usages
    private int mCount = 0;
    5 usages
    private TextView mtxtCount;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable( $this$enableEdgeToEdge: this);
        setContentView(R.layout.activity_main);
        mtxtCount = (TextView) findViewById(R.id.txtCount);
        if(mtxtCount !=null)
            mtxtCount.setText(Integer.toString(mCount));
        };
    1 usage
    public void showToast(View view){
        Toast toast = Toast.makeText( context: this,  text: "Hello Toast", Toast.LENGTH_SHORT);
        toast.show();
    }
    1 usage
    public void countUp(View view){
        mCount++;
        if (mtxtCount!=null)
            mtxtCount.setText(Integer.toString(mCount));
    }

    }
```


11:47
TOAST
1
1:1
Count


11:47
TOAST
1
1:1
Count