

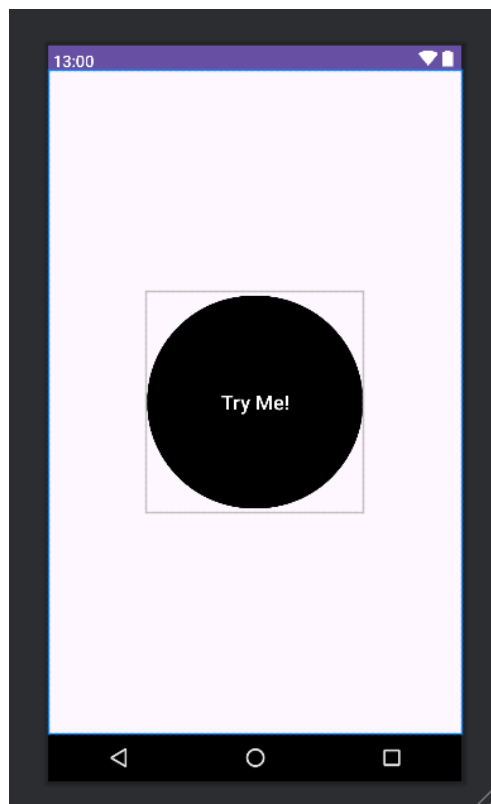
I. TryMeApp

- app này hoạt động là khi nhấn vào nút button thì sẽ background của nền sẽ thay đổi liên tục theo cơ chế random

II. Cách thực hiện

1. Design + property

- Design như sau:



- Property

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/windowViewID"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/tryMeButton"
        android:layout_width="215dp"
        android:layout_height="219dp"
        android:backgroundTint="@color/black"
        android:text="@string/try_me"
        android:textSize="20sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

2. Xử lý logic

```
public class MainActivity extends AppCompatActivity {

    private View windowView;
    private Button tryMeButton;
    private int[] colors;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // id, property
        windowView = findViewById(R.id.windowViewID);
        tryMeButton = findViewById(R.id.tryMeButton);
        colors = new int[] {Color.BLUE, Color.GREEN, Color.RED};

        // activity
        tryMeButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                int colorArrayLength = colors.length;
```

```

35
36         Random random = new Random();
37         int colorNum = random.nextInt(colorArrayLength);
38
39         windowView.setBackgroundColor(colors[colorNum]);
40
41         Log.d( tag: "Random", String.valueOf(colorNum));
42     }
43 }
44
45
46 }
47

```

+ ở dòng 28 thì chúng ta có thể sử dụng những class Color trong đó bởi vì do lớp Color có một phương thức getRGB() trả về giá trị kiểu int. Khi chúng ta thêm một đối tượng của lớp Color vào mảng colors, giá trị trả về của phương thức getRGB() sẽ được sử dụng làm giá trị của phần tử mảng.

+ Bên dưới hàm onClick thì chúng ta sẽ sử dụng class Random để có thể sử dụng random số nguyên thông qua method nextInt

+ Log.d là trạng thái debug của app sẽ trả về kết quả string của số nguyên colorNum