Add a UI control on the screen of the second activity so that you can close the second activity and go back to the first activity (i.e., the main activity). In addition, on the main activity, display an iteration count that reflects the number of times the main activity has been displayed on the front ground.

Main Activity XML: The Main Activity's given XML layout shows a top-left aligned Text View that reads "Iteration count: 0" The iteration count shows how many times the button is clicked and a button with the name "Launch Second Activity" that is horizontally centered and positioned 323dp from the top. The linked Java code launches a function called launchSecondActivity When the button is clicked it starts a second activity.

```
<?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:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

        <TextView
            android:id="@+id/iteration_count"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Iteration count: 0"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent" />

        <Button
            android:id="@+id/button"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="323dp"
            android:onClick="launchSecondActivity"
            android:text="Launch Second Activity"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

The second XML activity or the main activity2:in this activity "Close" is written on a button positioned in the center of "MainActivity2". The corresponding Java code, when activated, calls a method with the name "closeActivity," closing this activity and returning to the first one (the "MainActivity"), which has a button to launch the "MainActivity2" i.e. launch second activity and a "TextView" showing the number of iterations.

'MainActivity' can therefore launch 'MainActivity2', and 'MainActivity2' offers the choice to close itself and return to 'MainActivity'.

```
<?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:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity2">

        <Button
            android:layout_width="194dp"
            android:layout_height="60dp"
            android:onClick="closeActivity"
            android:text="Close"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintHorizontal_bias="0.5"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent"
            app:layout_constraintVertical_bias="0.5"
            tools:ignore="HardcodedText,MissingConstraints" />


</androidx.constraintlayout.widget.ConstraintLayout>
```

Main Activity.java code:

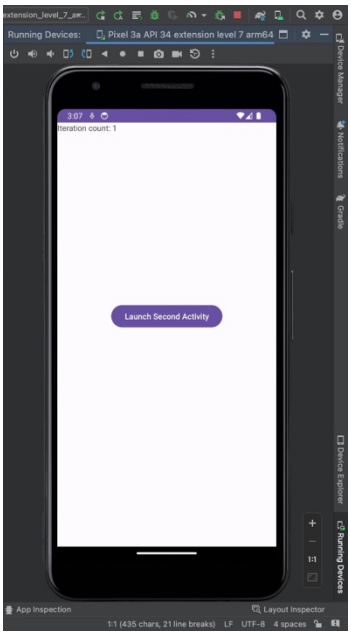
- The 'TextView' widget with ID iteration_count' is initialized to display the iteration count upon the establishment of activity, as defined by the 'onCreate' method, and the interface from R.layout.activity_main is set.
- The count increases when the activity is performed, as seen by the 'onResume' function.
- The private method 'updateIterationCount' controls this incrementing action and changes the widget's display.
- The activity also has a button that, when clicked, launches a second activity ('launchSecondActivity' method), which is linked via the 'android:onClick' attribute in the XML layout. This method navigates to the 'MainActivity2' activity using the 'Intent' class ('android.content.Intent').
- I used the 'View' class (from 'android.view.View') is also imported to provide a parameter type for the button's click callback, and the 'TextView' class (from 'android.widget.TextView') makes it easier to manipulate the text display showing the iteration.

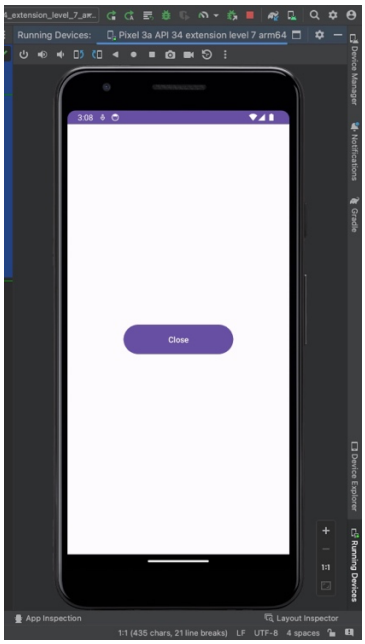
```java
package com.example.a2844629hw1;

import androidx.appcompat.app.AppCompatActivity;

import android.view.View;

import android.widget.TextView;

import android.content.Intent;

import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

        private int iterationCount = 0;
        private TextView iterationTextView;

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

            iterationTextView = findViewById(R.id.iteration_count);
            updateIterationCount();
        }

        @Override
        protected void onResume() {
            super.onResume();
            iterationCount++;
            updateIterationCount();
        }

        private void updateIterationCount() {
            iterationTextView.setText("Iteration count: " + iterationCount);
        }

        public void launchSecondActivity(View view) {
            Intent intent = new Intent(this, MainActivity2.class);
            startActivity(intent);
        }
    }
```

Main activity.java code :

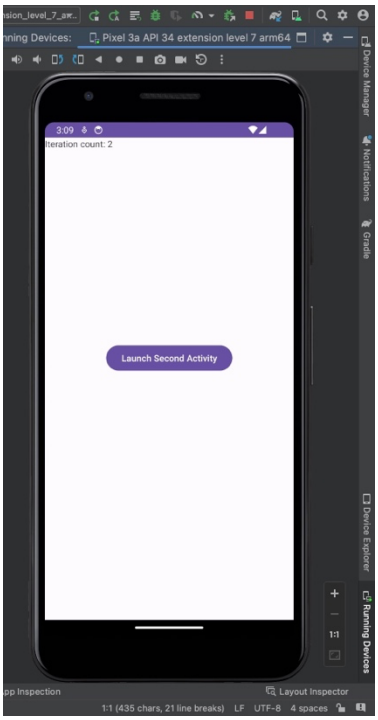
The 'MainActivity2' class extends 'AppCompatActivity' and is the second activity in the Android app. When launched, it uses the layout 'R.layout.activity_main2'. It has a 'closeActivity' function that, when called by a UI element click, uses the 'finish()' method to close the activity.

```java
package com.example.a2844629hw1;

import androidx.appcompat.app.AppCompatActivity;

import android.view.View;



import android.os.Bundle;

public class MainActivity2 extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);
    }

    public void closeActivity(View view) {
        finish();
    }
}
```

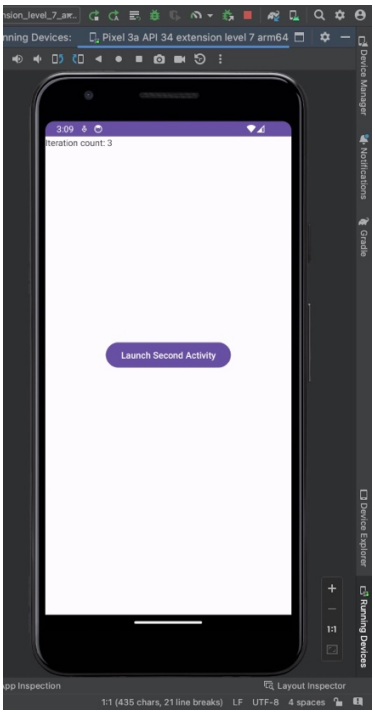




First screen (main activity) Second Activity



Back to main activity



Iteration count