

Objective

The objective of this practical work is to take a look at and understand a fundamental part of Android : **the activity lifecycle**. In Android, an activity is the entry point for interacting with the user. During its lifetime, an activity transitions through, and sometimes back to, various states. This transitioning of states is known as the activity lifecycle. The activity lifecycle extends from the creation of the activity to its destruction, when the system reclaims that activity's resources. As a user navigates away from and back to an activity, each activity transitions between different states in the activity lifecycle. In addition, the other objectives are to take a look at the **"Log"** class for writing messages from Android about app to the **"Logcat"** (**console for logging messages**), and the implementation of various lifecycle callback methods.

1. Practical Tasks

- (a) Create a new Android Project in Android Studio with a *Empty Activity*, and call it **TP_Activity_LifecycleGXXXX**.
- (b) Create a new Activity, and name it as **Second_Activity**.
- (c) Open **AndroidManifest.xml** and place `<meta-data>` balise on the second activity declaration for main activity come back.
- (d) Open **activity_main.xml** (UI of the m in activity) and add an *"Button"* View with text **open_activity2**.
- (e) Open *MainActivity.java* and :
 - i. Declar the static String variable *LOG_TAG*, above the class declaration, with the name of the underlying class using `getSimpleName()` method.
 - ii. examine the `onCreate()` method for this activity and modify it to write messages to the Logcat and handle the second activity starting event.
 - iii. Add and override all activity lifecycle method in Android to handle the remaining events.
- (f) Open *Second_Activity.java* and :
 - i. Declar the static String variable *LOG_TAG*, above the class declaration, with the name of the underlying class using `getSimpleName()` method.
 - ii. examine the `onCreate()` method for this activity and modify it to write messages to the Logcat using **" Log.d(LOG_TAG, " ");"** instruction.
 - iii. Add and override all activity lifecycle method in Android to handle the remaining events.
- (g) Compile and run the app. In Android Studio, at the bottom of the screen, click the **Logcat tab** and filter the logs to see the behavior differences in the app.

```

<activity android:name=".Second_Activity">
    <meta-data
        android:name="android.support.PARENT_ACTIVITY"
        android:value=".MainActivity" />
</activity>

```

FIGURE 1: Figure of AndroidManifest.xml

```

public class MainActivity extends AppCompatActivity {
    private static final String LOG_TAG = MainActivity.class.getSimpleName();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Log.d(LOG_TAG, "-----");
        Log.d(LOG_TAG, "Activity 1 invocation of callback onCreate");
        // écouteur du bouton bouton2
        Button btnsecond_activity = findViewById(R.id.button);
        btnsecond_activity.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // TODO treatment
                Intent intent = new Intent(getApplicationContext(),
                Second_Activity.class);
                startActivity(intent);
            }
        });
    }
    @Override
    public void onStart(){
        super.onStart();
        Log.d(LOG_TAG, "Activity 1 invocation of callback onStart");
    }

    @Override
    protected void onStop() {
        super.onStop();
        Log.d(LOG_TAG, "Activity 1 invocation of callback onStop");
    }

    @Override
    protected void onPause() {
        super.onPause();
        Log.d(LOG_TAG, "Activity 1 invocation of callback onPause");
    }

    @Override
    protected void onResume() {
        super.onResume();
        Log.d(LOG_TAG, "Activity 1 invocation of callback onResume");
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        Log.d(LOG_TAG, "Activity 1 invocation of callback onDestroy");
    }
}

```

FIGURE 2: Figure of MainActivity.java

```

public class Second_Activity extends AppCompatActivity {
    private static final String LOG_TAG = Second_Activity.class.getSimpleName();
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_second_);
        Log.d(LOG_TAG, "-----");
        Log.d(LOG_TAG, "Activity 2 invocation of callback onCreate");
    }
    @Override
    public void onStart(){
        super.onStart();
        Log.d(LOG_TAG, "Activity 2 invocation of callback onStart");
    }

    @Override
    protected void onStop() {
        super.onStop();
        Log.d(LOG_TAG, "Activity 2 invocation of callback onStop");
    }

    @Override
    protected void onPause() {
        super.onPause();
        Log.d(LOG_TAG, "Activity 2 invocation of callback onPause");
    }

    @Override
    protected void onResume() {
        super.onResume();
        Log.d(LOG_TAG, "Activity 2 invocation of callback onResume");
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        Log.d(LOG_TAG, "Activity 2 invocation of callback onDestroy");
    }
}

```

FIGURE 3: Figure of Second_Activity.java