UNIVERSIDADE DA CORUÑA

Systems Programming 20/21 Q1

FractalsApp

A Learning Mobile Application for Fractals

Author: Beatrix-Valeria Csaki

Email: b.csaki@udc.es

Date: 27 January 2021

Contents

1 Introduction

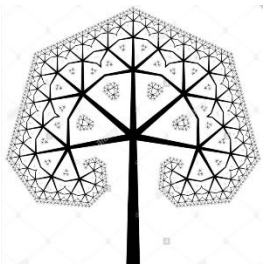
Fractals are not just complex shapes and pretty pictures generated by computers. Anything that appears random and irregular can be a fractal. They permeate our lives, appearing in places as tiny as the membrane of a cell and as majestic as the solar system. Fractals are the unique, irregular patterns left behind by the unpredictable movements of the chaotic world at work.

Therefore, since I found the field of fractals interesting, I wanted to learn more about this subject and its applications in real life and also creating a mobile application seemed to be appropriate and challenging in the meanwhile.

Fractals have more and more applications in science. The main reason is that they very often describe the real world better than traditional mathematics and physics. They have applications in fields like:

- astronomy (eg: structure of the universe),
- nature (eg: trees),
- fluid mechanics (eg: study of turbulence in flows),
- telecommunication (eg: fractal-shaped antennae that reduce greatly the size and the weight of the antennas),
- physics (eg: the roughness of surfaces),
- medicine (eg: biosensor interactions),
- computer science (eg: image compression)

Actually, the most useful use of fractals in computer science is the fractal image compression. This kind of compression uses the fact that the real world is well described by fractal geometry. By this way, images are compressed much more than by usual ways (eg: JPEG or GIF file formats). Another advantage of fractal compression is that when the picture is enlarged, there is no pixelization. The picture seems very often better when its size is increased.



1

2

[1] Computer generated tree with fractal shapes. URL: https://www.alamy.com/stock-image-computer-generated-tree-with-fractal-shapes-163130705.html

[2] Particles of abstract fractal forms on the subject of nuclear physics science and graphic design URL: https://www.123rf.com/photo_128841127_particles-of-abstract-fractal-forms-on-the-subject-of-nuclear-physics-science-and-graphic-design-geo.html

1.1 Objectives

FractalsApp is a learning mobile application that introduces the user to the universe of fractals. The main objective of the presented application is to provide a learning platform and the possibility to check the learned skill, as a secondary one.

1.2 Motivation

In my opinion, there are a lot of students that would find fractals interesting and most of the applications are not really designed to teach students about fractals. They are usually made to generate and let the user to create beautiful fractal images. Even if they provide a fun playground for this kind of experiments, I think that an application which combines both learning and playing is a good idea.

Therefore, my application is structured in two parts. The first part is a tutorial which provides the user general information about fractals (what they are, real life applications, etc) and it will be structured by chapters. There is the second part for testing the knowledge that the user gained from the first part, which will consist of interactive exercises and quizzes that will generate a score to see how well the user understood the presented information. It will also help the user maintain a relaxed attitude during the learning process.

1.3 Related work

There are several applications regarding fractals already existing on Google Play Store platform.

”Fractview”[3] is a viewer for fractals and complex functions with many features. The app supports very intuitive gestures like pinch-to-zoom (moving, rotation, scaling). The philosophy is similar to Google Maps - moving a finger moves the point below it accordingly. It also contains presets for Newton fractals, Orbit traps, even pendulum simulations[4] which allow the user so easily enter custom functions. There is an intuitive editor for (2-dimensional) color palettes. Images can be saved (as pngs) and shared. Furthermore, the user can save the fractal description and send it to others to share their work.

”Fract”[5] is a fast, highly customizable, open source, free fractal generator. The aim of this program is to create artistic fractal images. It provides a list of demos with a large list of fractal formulas and drawing methods. The fractals are calculated using an internal, Kotlin-alike programming language.

[3] URL: https://play.google.com/store/apps/details?id=at.searles.fractview
[4] URL: https://nylander.wordpress.com/2007/10/27/magnetic-pendulum-strange-attractor/
[5] URL: https://play.google.com/store/apps/details?id=at.searles.fract

2 Requirements analysis

2.1 Functionalities

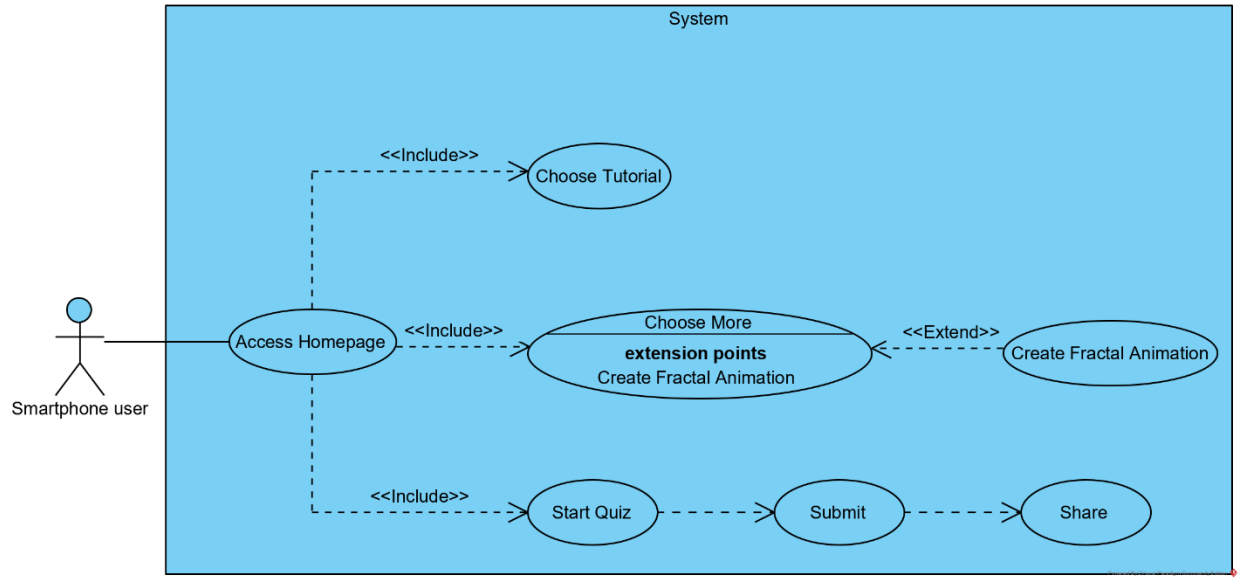


Figure 1: Use Case Diagram

The homepage will consist of 3 buttons:

- First one will link to the learning part, divided by chapters
- The second one will link to a WebView Interface[6] which provides further/related information (for example an URL which shows a tutorial on how to create a fractal animation in JavaScript using the Julia fractal)
- Third one consists the quiz:
  - The result of the quiz will be shown in the app with a button, which shows the answer in a ToastView Interface[7]
  - And another button for sharing the answer on Facebook

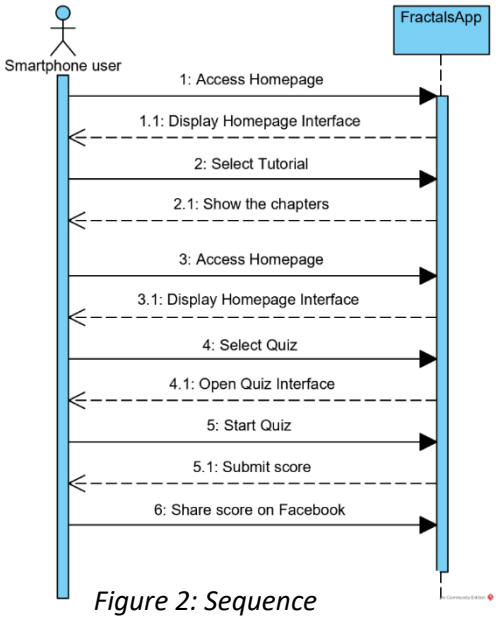


Figure 2: Sequence Diagram at System Level

[6] an extension of Android's View class that displays web pages as a part of the activity layout
[7] a view containing a quick little message for the user Interface.

2.2 GUI Prototype

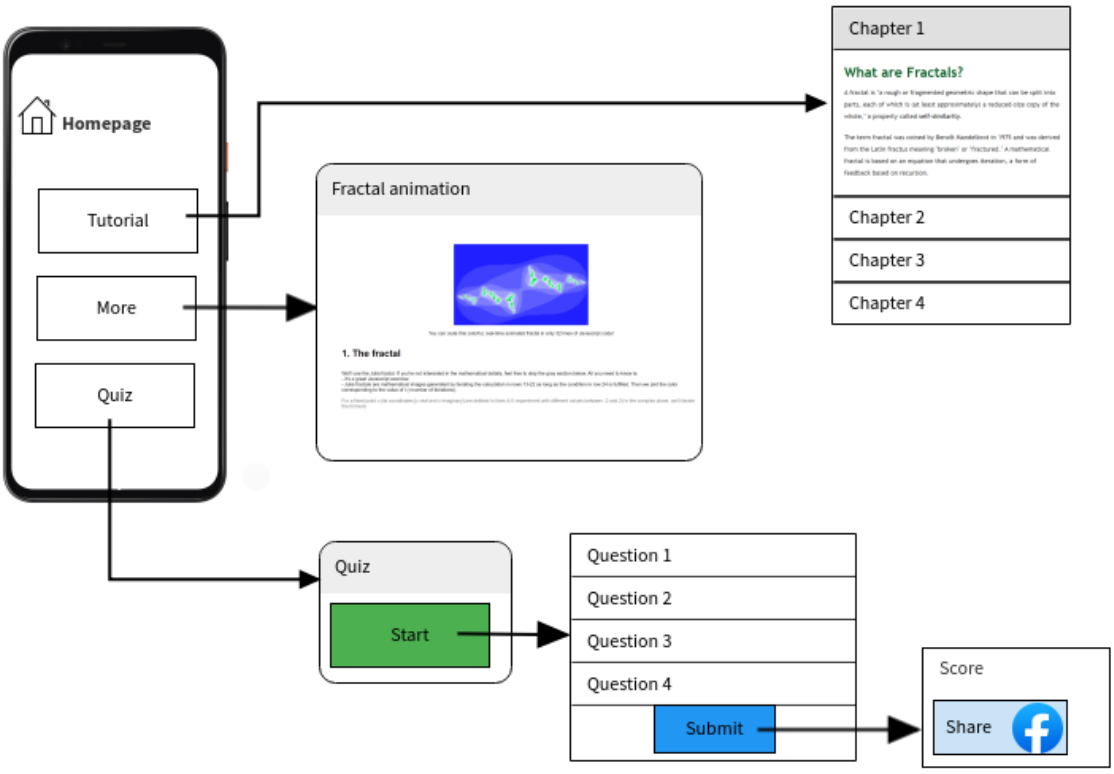


Figure 3: Graphic User Interface Prototype

3 Planning

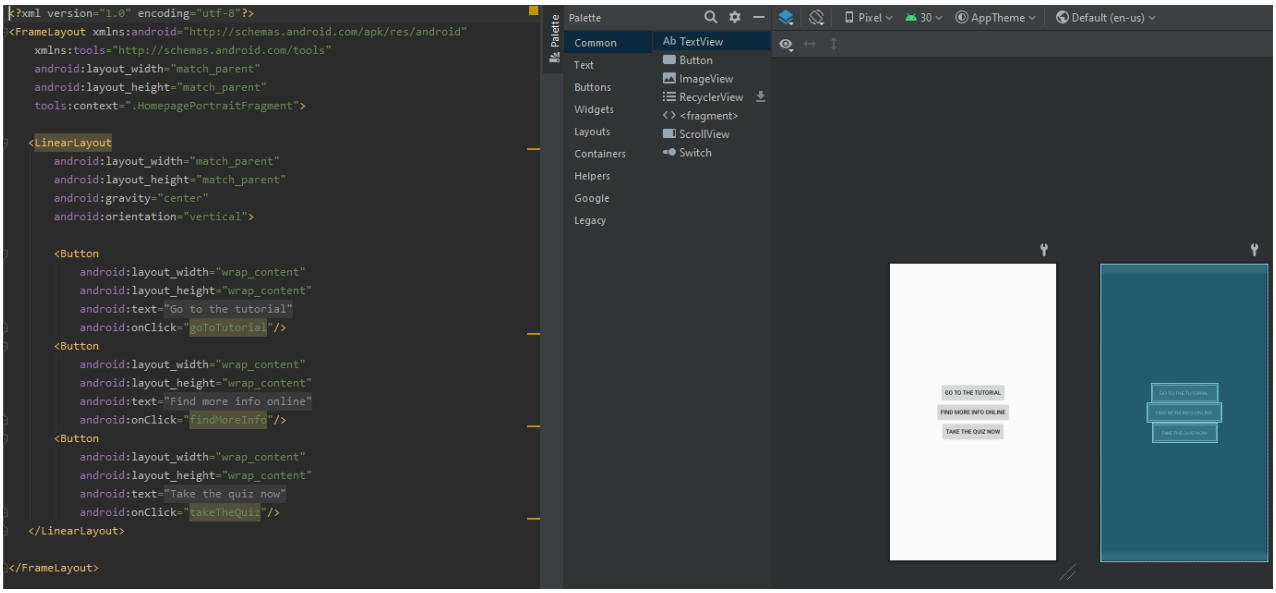
3.1 Iterations

The order wants to be as follows: First design the home activity that will contain the buttons from which the user will choose what to do. Those buttons are: “Go to tutorial” that opens the tutorial activity which contains information. “Find more info online” that opens an activity which will contain a WebView of a link containing facts about fractals and “Take the quiz now” which opens the quiz activity in order to asses what the user has learned so far.

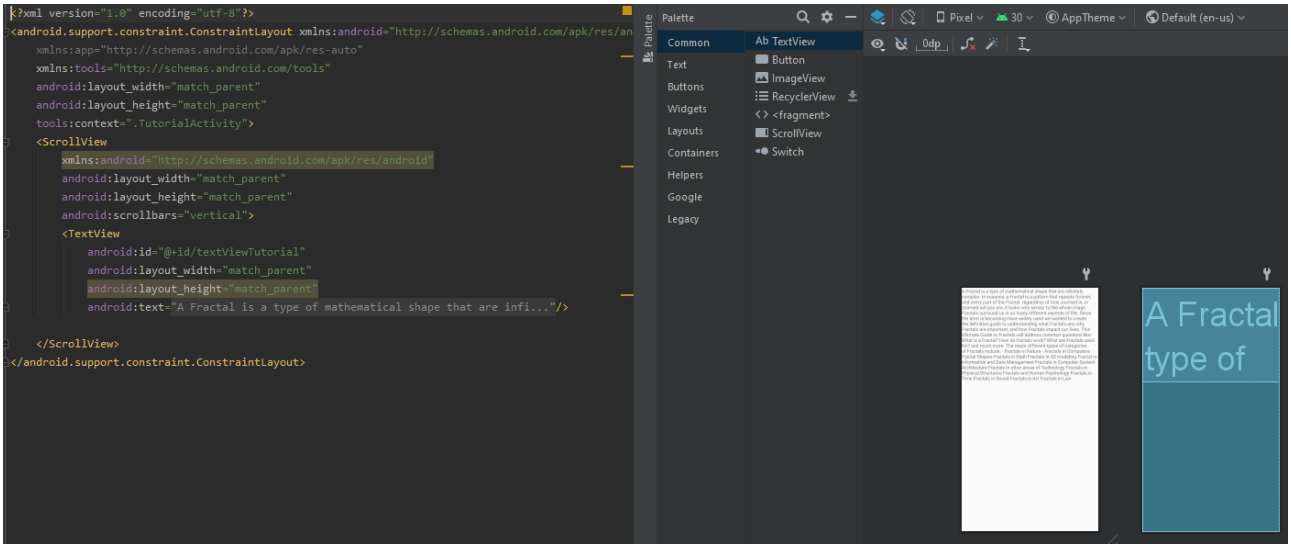
3.2 Milestones and deliverables

- Design and implement the Main Activity
- Design and implement the WebView for the activity and link it to the Main Activity through a button
- Design and implement the quiz activity that and create the quiz

4 Design



This is the layout of the Main Activity which contains the three buttons in the center that links the user to the other activities.

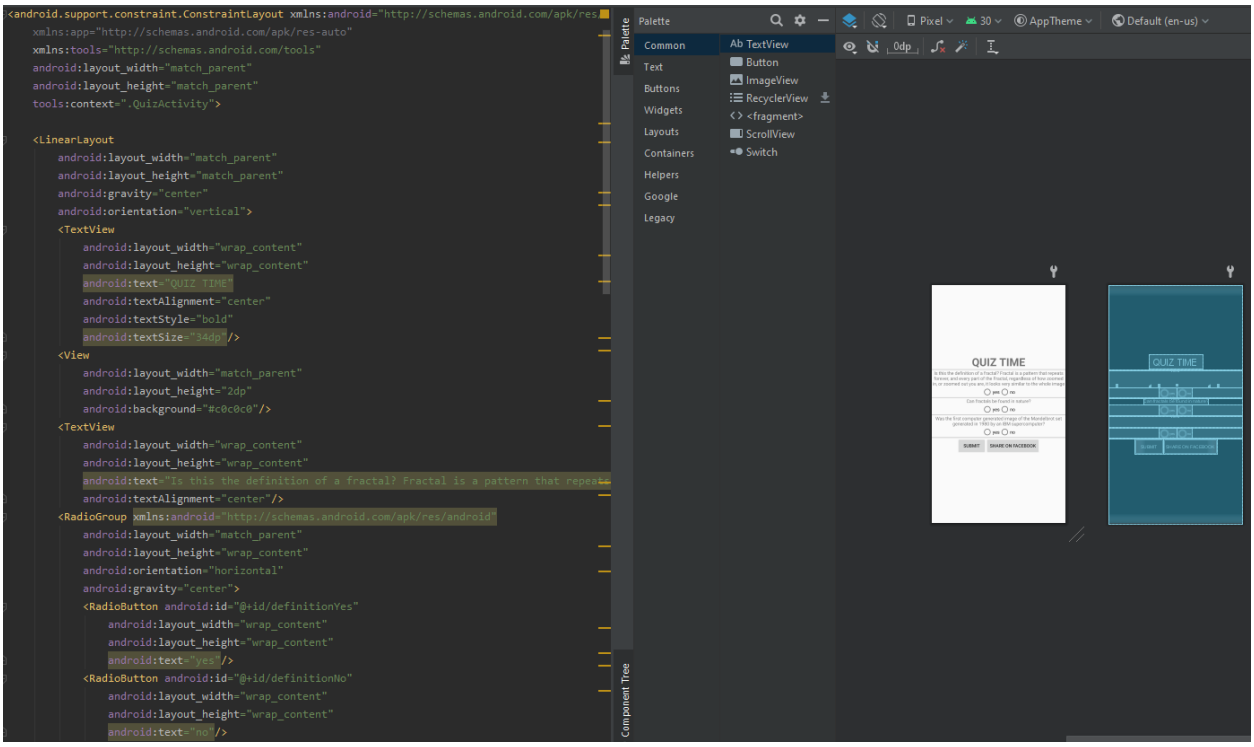


This is how the Tutorial Activity looks like, just a Scroll View that acts as a container for the information provided to the user.

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.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=".ViewMoreWebviewActivity">

    <WebView
        android:id="@+id/webview"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        />
</android.support.constraint.ConstraintLayout>
```

The View more activity is all a Web View where the link is loaded and displayed to the user.



The Quizz Activity consists of a series of Views for the text of the questions and Radio buttons for the answers.

4.1 Architecture

The application consists of 4 simple activities:

- MainActivity, which contains 2 fragments:
  - HomepageLandscapeFragment
  - HomepagePortraitFragment
- TutorialActivity
- ViewMoreWebviewActivity
- QuizActivity

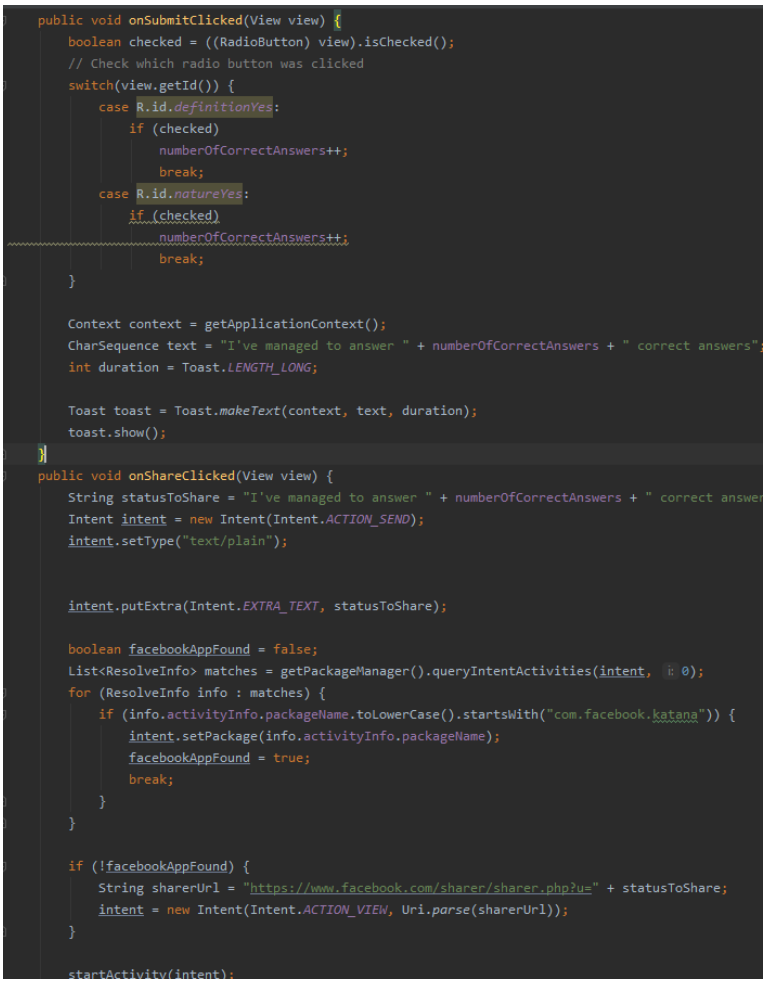
The homepage contains 3 buttons, using button lists.

```
public void goToTutorial(View view) {
    Intent intent = new Intent(getContext(), TutorialActivity.class);
    startActivity(intent);
}
public void findMoreInfo(View view) {
    Intent intent = new Intent(getContext(), ViewMoreWebviewActivity.class);
    startActivity(intent);
}
public void takeTheQuiz(View view) {
    Intent intent = new Intent(getContext(), QuizActivity.class);
    startActivity(intent);
}
```

Tutorial Activity is launched from the Intent correspondent to the Button from Main Activity then the layout is loaded and the Scroll View displayed.

View more on web activity is launched from Main Activity and the layout for the Web View is loaded and displayed.

Quizz Activity is also launched in the same way but it also contains two methods responsible for handling Submit and Share actions, as seen below.



```
public void onSubmitClicked(View view) {
    boolean checked = ((RadioButton) view).isChecked();
    // Check which radio button was clicked
    switch(view.getId()) {
        case R.id.definitionYes:
            if (checked)
                numberOfCorrectAnswers++;
                break;
        case R.id.natureYes:
            if (checked)
                numberOfCorrectAnswers++;
                break;
    }

    Context context = getApplicationContext();
    CharSequence text = "I've managed to answer " + numberOfCorrectAnswers + " correct answers";
    int duration = Toast.LENGTH_LONG;

    Toast toast = Toast.makeText(context, text, duration);
    toast.show();
}
public void onShareClicked(View view) {
    String statusToShare = "I've managed to answer " + numberOfCorrectAnswers + " correct answer
    Intent intent = new Intent(Intent.ACTION_SEND);
    intent.setType("text/plain");


    intent.putExtra(Intent.EXTRA_TEXT, statusToShare);

    boolean facebookAppFound = false;
    List<ResolveInfo> matches = getPackageManager().queryIntentActivities(intent,  0);
    for (ResolveInfo info : matches) {
        if (info.activityInfo.packageName.toLowerCase().startsWith("com.facebook.katana")) {
            intent.setPackage(info.activityInfo.packageName);
            facebookAppFound = true;
            break;
        }
    }

    if (!facebookAppFound) {
        String sharerUrl = "https://www.facebook.com/sharer/sharer.php?u=" + statusToShare;
        intent = new Intent(Intent.ACTION_VIEW, Uri.parse(sharerUrl));
    }

    startActivity(intent);
```

4.2 Persistence

The data will be stored inside the application, in strings.xml file and will be managed through the strings name as seen below.

```
<string name="app_name">FractalsApp</string>
<string name="hello_blank_fragment">Hello blank fragment</string>
<string name="go_to_tutorial">Go to the tutorial</string>
<string name="find_info_online">Find more info online</string>
<string name="take_the_quiz">Take the quiz now</string>
```

4.3 View

Each activity is composed by a list of graphics components and these components link one activity to another through buttons that create Intents in order to handle the activity/fragment transitions.

```
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
                         Bundle savedInstanceState) {
    // Inflate the layout for this fragment
    return inflater.inflate(R.layout.fragment_homepage_portrait, container,  attachToRoot: false);
}
```

Method *onCreateView* handle the layout loading for fragments when they are triggered by Intents.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_tutorial);
    TextView helloTextView =  findViewById(R.id.textViewTutorial);
    helloTextView.setMovementMethod(new ScrollingMovementMethod());
}
```

Method *onCreate* handle the layout loading for activities when they are triggered by Intents.

4.4 Communications

The data between activities is passed through Intents. To retrieve the data, it is used the method *getExtra()* on the next activity from the Intent. But there is not data needed to pass between fragments or activities.

The only time when data needs to be passed is when the user chooses to share the quiz result on Facebook and then the following method gets the Facebook application installed on the device and launches it in order to share the result.

```
public void onShareClicked(View view) {
    String statusToShare = "I've managed to answer " + numberOfCorrectAnswers + " correct answers";
    Intent intent = new Intent(Intent.ACTION_SEND);
    intent.setType("text/plain");


    intent.putExtra(Intent.EXTRA_TEXT, statusToShare);

    boolean facebookAppFound = false;
    List<ResolveInfo> matches = getPackageManager().queryIntentActivities(intent,  0);
    for (ResolveInfo info : matches) {
        if (info.activityInfo.packageName.toLowerCase().startsWith("com.facebook.katana")) {
            intent.setPackage(info.activityInfo.packageName);
            facebookAppFound = true;
            break;
        }
    }

    if (!facebookAppFound) {
        String sharerUrl = "https://www.facebook.com/sharer/sharer.php?u=" + statusToShare;
        intent = new Intent(Intent.ACTION_VIEW, Uri.parse(sharerUrl));
    }

    startActivity(intent);
}
```

4.5 Sensors

The application uses orientation sensors which measure the physical position of the device. This leads to two layouts landscape and portrait.

```
if (config.orientation == Configuration.ORIENTATION_LANDSCAPE) {


    android.support.v4.app.FragmentTransaction ft = getSupportFragmentManager().beginTransaction();
    ft.replace(R.id.fragment, new HomepageLandscapeFragment());
    ft.commit();
}else{
    android.support.v4.app.FragmentTransaction ft = getSupportFragmentManager().beginTransaction();
    ft.replace(R.id.fragment, new HomepagePortraitFragment());
    ft.commit();
}
```

When the Main Activity is launched this if checks the orientation of the device and launches the correspondent activity, depending on orientation the portrait or the landscape layout must be loaded.

4.6 Work in background

The score calculation of the quiz. I did not use a service because it is not necessary for this case.

5 Bibliographic references

- https://developer.android.com/
- https://fractalfoundation.org/