



MATERIAL DESIGN

Android Elective Course 4th Semester

June 2016

Teacher: Anders Kristian Børjesson

Ovidiu Floca

1 CONTENTS

2	Introduction	2
3	Problem Definition	2
4	Method	2
5	Planning.....	3
6	Watching the keynotes from Google I/O	3
7	Reading Google’s guide for making a material design app.....	4
8	Making a material design application	6
9	Conclusion.....	9
10	References	10

Material Design

2 INTRODUCTION

In the following report I want to discuss the Android Material Design libraries, with its elements, uses and features. Material Design is a new set of design rules with the purpose to improve the user experience and to modernize the way applications look.

The reason I picked this subject is that I am interested in the way I can use Material Design in my applications, because it's the standard that should be used in all new Android applications. This new methodology can be useful as it provides a way to make the experience of moving from one Android device from one manufacturer to a completely different one completely seamless.

3 PROBLEM DEFINITION

I choose 3 questions which I will use to guide me in finding more about the new Material Design methodology

1. What defines Material Design?
2. How to use Material Design?
3. How useful is Material Design?

The answer for the first question should explain more about this new methodology from Google.

The second one should show why Google is using it and why they think that it's better than Halo, the old design methodology. It should also show the difference from the developer's point of view of implementing the new changes.

The third one aims to look at the design methodology from the user's point of view and analyze if Material Design is in any way more useful for the normal user.

4 METHOD

For finishing the research, I choose 3 activities that should answer the previous questions:

1. Watching the keynotes from Google I/O¹. The purpose of this activity is to get Google's point of view regarding Material Design and also to find out more about how they made it and what they plan to do with it.
2. Reading Google's guide for making a Material Design application. For this activity I will go to the developer website in order to read about and understand the ways to implement Material Design in applications, which will prepare me for the third activity.

¹ <https://www.youtube.com/watch?v=isYZXwaP3Q4>

3. Making a Material Design application. The best way to find about Material Design and how to use it is by making an application.

5 PLANNING

The time budget for this small research is 5 weeks, while the page budget is 10 pages.

In the first week I will be watching the Keynotes from Google I/O 2014 and write about what I see there and about Google's opinion about this.

In the second week, I am planning to read and understand from the Android developer website and I will write about that.

In the third and fourth weeks, I will make an Android application that uses Material Design elements and I will document it.

In the last week, I will make sure that everything is in order with the report and I will fix the problems that may arise.

6 WATCHING THE KEYNOTES FROM GOOGLE I/O

Google, the owner of Android, released the Material Design philosophy in 2014 at Google I/O, the annual Google conference where their most important people talk about new technologies.

Material Design was presented by Google as a new design methodology with the purpose to make using the user interface a much simpler task for the user by presenting everything in a natural and intuitive way.

The creators tried to make the user interface physical by adding life-like shadows and edges and also by having the content displayed into "intelligent paper".

The core of Material design are cards that give the impression that they are made out of paper, but unlike real paper it can expand and rearrange itself intelligently making the interface "easier to understand than language" according to Google's Vice President of Design Matias Duarte. The reason behind this statement is that the physical movements and the tangible surfaces presented inside the user interface are made to look like in real life by using the physical properties of paper and light. The designers behind it spent hours putting pieces of papers on top of each other and bending them under light at different angles to find the perfect combination and they transposed that on a 2 dimensional screen making it feel 3 dimensional. The card in action can be seen here: <https://goo.gl/fSOJWp>

Unlike plastic buttons, the buttons presented in material design give the feeling that they rise towards the finger and touching it when you press it. The actual spot where the content is displayed on the screen is around 0.5-1mm under the glass that protects it. By making the buttons rise towards the finger, the designers wanted to create the illusion that the button actually is touching the finger regardless of the protective glass over it. The previous statement can be seen in action here: <https://goo.gl/Hm2z9Q>

Animations are another point of interest for Material Design. The center of the animation is the user input and the reason for that is because almost every change that happens inside an application is caused by the user. The animations are used to give feedback to the user. The animations show continuity from one item to the other by expansion or by sliding, the same way a piece of paper slides over another. With the help of Material Design the user is more aware of the changes that happen from one state to the other. The user knows and understands very easy how to go back to the previous state and also what happened to get to the current one. The animations are used to direct the focus of the user to the important stuff that is happening inside the application.

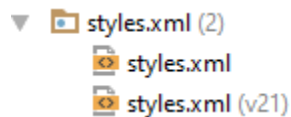
7 READING GOOGLE'S GUIDE FOR MAKING A MATERIAL DESIGN APP

There are a few questions a developer should ask himself/herself before starting to develop with Material Design:

1. Will it be compatible with all the devices that I want to support?
2. Is it hard to develop with it?

I started to look for the answers to the previous questions on the official Android developer website and I found out that there are multiple ways to provide support for Android versions older than 5.0.

One way is to set the material theme for Android versions newer than 5.0 and to use older themes (like Holo) for previous versions. This is done by having 2 separate "styles.xml" files as in the following picture:



The "styles.xml (v21)" is for Android 5 (API 21).

Another way to have compatibility across versions is to use different layout files for API 21 and older. This is done similarly to the procedure described above.

In my opinion, the best way to use Material Design with older androids is using the "v7 support libraries". These are libraries designed to be used with Android 2.1 or higher (API 7) and provide a wide array of features from newer Android that can be used on older ones. In order to access the design features from Android 5 the latest versions of the following dependencies must be added in Gradle:

```
dependencies {
    compile 'com.android.support:appcompat-v7:23.3.0'
    compile 'com.android.support:cardview-v7:23.3.0'
    compile 'com.android.support:recyclerview-v7:23.3.0'
}
```

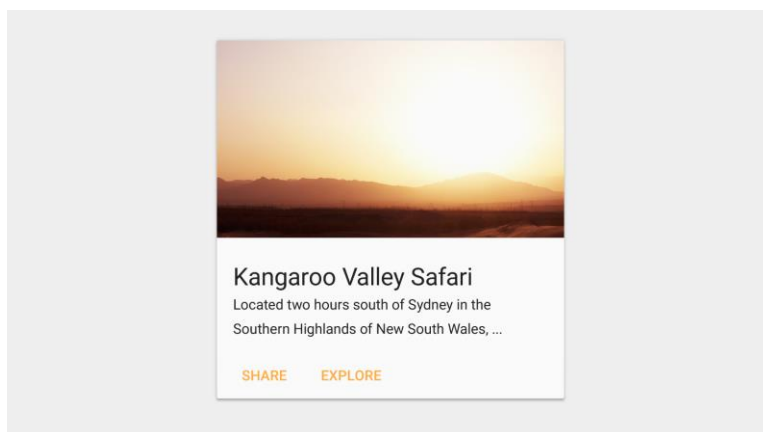
The new Material Design libraries provide a lot of new things that can be done to the application, but that also mean that the developer has to learn those things and understand them. One of those things is the RecyclerView², which is an improved version of ListView. The RecyclerView is used to display large data sets, like the ListView.

One of the center items of Material Design is the CardView. It lets you show information inside card like layouts and they all look the same no matter what app you are using. The following code shows how to declare a CardView inside a layout:

```
<android.support.v7.widget.CardView
    xmlns:card_view="http://schemas.android.com/apk/res-auto"
    android:id="@+id/card_view"
    android:layout_gravity="center"
    android:layout_width="300dp"
    android:layout_height="250dp"
    card_view:cardCornerRadius="3dp">

    <TextView
        android:id="@+id/info_text"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
</android.support.v7.widget.CardView>
```

The Material Design card must be only about one subject, it can display links, text, pictures about this subject. Cards must have rounded corners :



² <https://developer.android.com/reference/android/support/v7/widget/RecyclerView.html>

Another material feature is the ability to cast shadow in order to define the height of an item. This can be done using the `View.setElevation()` method or when declaring an item inside the layout file by adding the property: `android:elevation`.

8 MAKING A MATERIAL DESIGN APPLICATION

In my quest for finding out more about Material Design and the way to implement it inside an Android application, I decided to create a new app which shows the new animations, the card view and also other Material Design elements like the new button designs and the loading spinner. The application also makes use of the `AppCompatActivity` library in order to be compatible with Android versions below 5.0, bringing the Material Design look to old devices.

Most of the work for implementing Material Design is done in the `.xml` layout files. In the activity classes, the programming part is done similarly. `OnClickListener` is implemented identical as in older versions of Android, even for the new elements like `CardView`.

I started by creating a new project in Android Studio with one Activity. There are options to make activities with material elements, like a Navigation Drawer, or Floating action buttons and the layout files and the `MainActivity` is generated automatically with all the necessary code and properties.

The theme is defined in the `styles.xml`. Here, the theme properties are set like in the example below

```
<style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
    <item name="colorPrimary">@color/colorPrimary</item>
    <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
    <item name="colorAccent">@color/colorAccent</item>
</style>

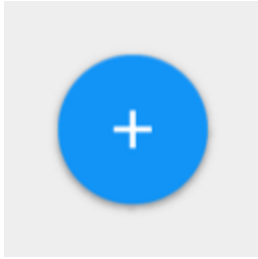
<style name="AppTheme.NoActionBar">
    <item name="windowActionBar">>false</item>
    <item name="windowNoTitle">>true</item>
</style>

<style name="AppTheme.AppBarOverlay" parent="ThemeOverlay.AppCompat.Dark.ActionBar" />
<style name="AppTheme.PopupOverlay" parent="ThemeOverlay.AppCompat.Light" />
```

The `AppCompatActivity.Light.DarkActionBar` theme is used, which is based on the latest Design style set by Google for Android (in this case Material Design). The app colors are set here and other app properties.

The rest of the elements (`RadioButton`, `ProgressBar`, `Slider` etc.) are declared normally in the `layout.xml` file and they will be rendered with the selected theme, which would be the material theme.

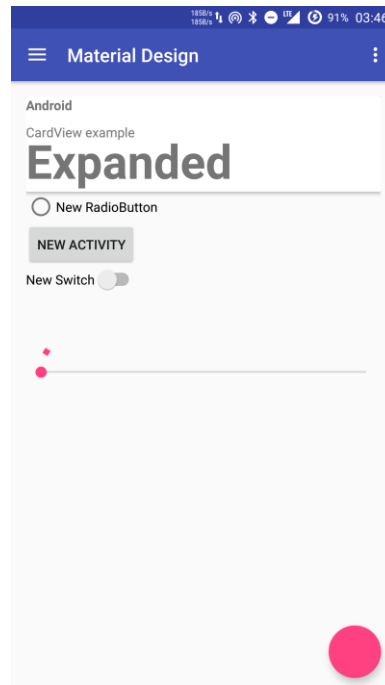
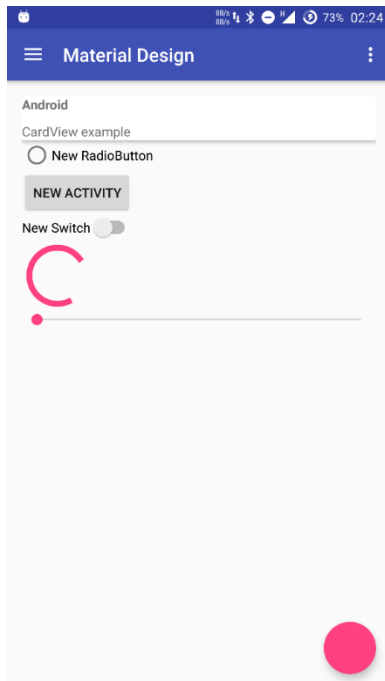
The floating action button, new type of button introduced in Android 5.0 with the Material Theme.



It is declared similarly to other Buttons inside the layout.xml file using the **android.support.design.widget.FloatingActionButton** tag, which offers a similar button for older Android devices. Any floating action button can be declared and manipulated inside the activity class like in the example below:

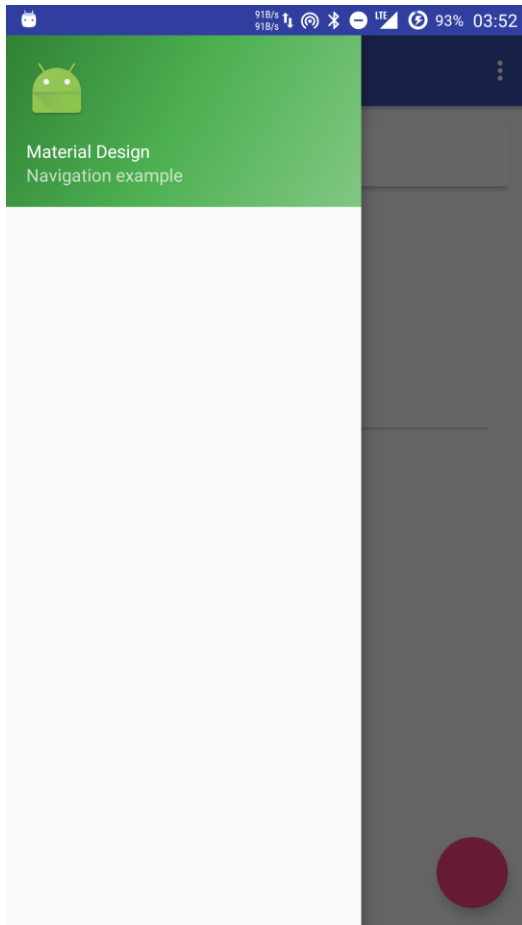
```
FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
fab.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        tv.setTextSize(50);
        tv.setVisibility(View.VISIBLE);
    }
});
```

In this case, the Floating Action Button makes a TextView present inside the CardView appear, making the CardView expand in order to showcase the new Material animations and the way the card grows with the content to accommodate it. The content below the card also moves with the card



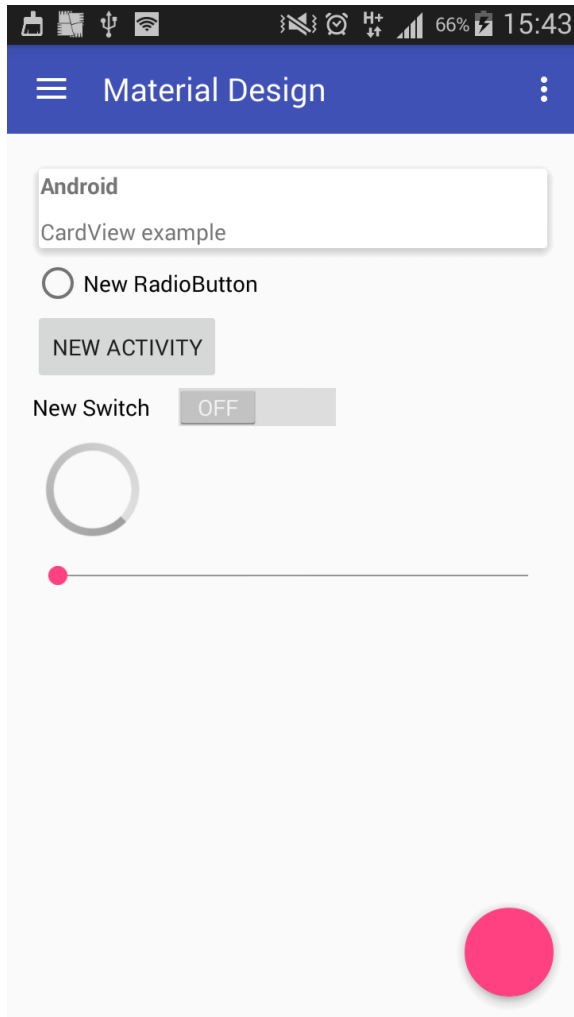
I also added a Navigation Drawer to the app in order to show the “paper over paper” design where the activities look like two pieces of paper on top of each other. It can be accessed by sliding from the left side of the screen or by pressing the “hamburger”³ button, next to the app title, in the upper right corner.

The navigation drawer gives the impression that it is on top of the activity by casting a shadow over it. The intensity of the shadow increases based on how much of the drawer is over the activity.



The Appcompat library does a good job of keeping the older versions of Android up to date with the new Design methodology.

³ <http://www.androidpolice.com/2014/10/30/google-turns-design-inconsistency-ten-latest-round-navigation-drawers/>



The previous screenshot is from a pre-Lollipop Android device.

The new Material Design elements introduced in Android 5 (the floating action button, the card view) are working as expected. The shadows and the animations for when the button is pressed are identical with the ones from a newer device, even the “hamburger” button works as expected.

Not all the elements are identical though. For example, the button, radio button and checker are displayed using the Holo design methodology, the one that Material Design replaced.

All in all, the AppCompat library did a good job of making the previous and current Android versions look and behave similarly. Other factors can be responsible for the way the applications look on both older and newer Androids.

9 CONCLUSION

What defines Material Design?

After the research done here I can say that Material Design can be defined by some predefined rules inspired from real life, the purpose of the content that is supposed to be displayed and imagination of

the person that is using it. The Material Design and all of its elements (card view, floating action button, etc.) can be applied to most of the “Holo” apps in order to give them a more modern look.

How to use Material Design?

The developers and designers behind Material Design made the use of this methodology inside an application easy, by having it well documented and with plenty of examples on the Android Developer website. It is implemented in a similar way as in Android KitKat or older in both the layout.xml files and the activity classes.

During the time that I took to research Material Design, I got to understand ways to make my future application more user friendly, more intuitive and modern looking. The same Material Design methodology can be applied for websites also, Google providing libraries for web development^{4 5 6}.

At the exam I intend to give an answer to the final question by demonstrating the same application work on two devices: one that supports Material Design only with the AppCompat library and another one that supports it natively.

10 REFERENCES

<https://developer.android.com/design/material/index.html>

<https://www.google.com/design/spec/material-design/introduction.html>

<https://developer.android.com/training/material/index.html>

<https://www.youtube.com/watch?v=isYZXwaP3Q4>

⁴ <https://getmdl.io/>

⁵ <https://elements.polymer-project.org/browse?package=paper-elements>

⁶ <https://material.angularjs.org/latest/>