

Konstantin Bork

Entwicklung eines Dortmund- Fremdenführers für Android-Smartphones

Heisenberg-Gymnasium Dortmund

GK Informatik 12

Betreuender Lehrer: 

Inhaltsverzeichnis

1	Einleitung	3
1.1	Geschichte von Android	3
1.2	Grundlagen für Android	4
1.2.1	Benötigte Kenntnisse	4
1.2.2	Benötigte Komponenten	5
1.3	Vorteile gegenüber anderen Systemen	5
2	Hauptteil	6
2.1	Die Idee zur App	6
2.2	Die Modellierungsphase	6
2.3	Die Bestandteile	6
2.3.1	Die Startseite	7
2.3.2	Das Auswahlfenster	8
2.3.3	Eine Beispiel-Sehenswürdigkeit	9
3	Fazit	11
3.1	Lernergebnisse	11
3.2	Mögliche Erweiterungen der App	11
3.3	Mögliche Entwicklung von Android	12
4	Anhang	13
4.1	Statistiken	13
4.2	Grafiken	13
4.3	Quelltexte	15
4.3.1	startActivity.java	15
4.3.2	main.xml	16
4.3.3	menu.java	16
4.3.4	menu.xml	19
4.3.5	Sight.java	20
4.3.6	sight.xml	22
4.3.7	strings.xml	22
4.3.8	AndroidManifest.xml	23
	Literatur	24
5	Eidesstattliche Erklärung	27

1 Einleitung

Im Jahr 2010 wurden laut BITKOM rund 357 Millionen € mit Apps für Smartphones und Tablet-PCs umgesetzt. Der Umsatz stieg dabei um 88% gegenüber dem Vorjahr. Für die nächsten Jahre ist aufgrund des steigenden Absatzes von Smartphones ein weiter zunehmender Umsatz vorhergesagt¹. Daher wird die Entwicklung für Smartphones zunehmend lukrativer.

Diese Facharbeit beschäftigt sich mit der Entwicklung einer Beispielanwendung für Android, einem Smartphone-Betriebssystem von Google. Anhand der vorgestellten, prototypisch umgesetzten Applikation „Dortmund Guide“ soll ein kleiner Einstieg in die App-Entwicklung für die Android-Plattform geschaffen werden.

1.1 Geschichte von Android

Im Jahr 2005 übernahm Google das auf die Entwicklung von Anwendungen für mobile Endgeräte spezialisierte Unternehmen Android Inc., wodurch „erste Gerüchte aufkamen, dass das Unternehmen in den Mobilfunkmarkt einsteigen wollte“². Nach langer Stille gab Google im Jahr 2007 die Gründung der „Open Handset Alliance“ bekannt, einem Konsortium, dem mittlerweile „65 Firmen aus dem Technologie- und Mobile-Bereich angehören“³. Sie unterstützt Google bei der Weiterentwicklung von Android. Im selben Jahr kündigte Apple die Veröffentlichung des iPhone an. Der Marktstart dieses Handys läutete einen Wandel ein, „der das Mobiltelefon in einen tragbaren, internetfähigen Computer verwandelt“⁴. „Bis das erst Android-Smartphone auf den Markt kam, dauerte es aber noch einige Zeit“⁵. Zunächst veröffentlichte Google „im Herbst 2008 den Quellcode von Android unter einer Mischung verschiedener Open-Source-Lizenzen“⁶, kurz darauf erschien dann das T-Mobile G1 als erstes Endgerät mit Android⁷. Seitdem erschienen bislang sieben Versionen, die „neben ihrer Versionsnummer einen englischsprachigen Spitznamen, der eine Süßspeise darstellt“⁸, haben. Mit der

¹vgl. [Ris 6]

²[Kö 6]

³[Gla10, S.18]

⁴[BP09, S. v]

⁵[Dir 6]

⁶[Kö 6]

⁷vgl. [Dir 6]

⁸[Dir 6]

Version 2.3 (Gingerbread) und der Version 3.0 (Honeycomb) fand eine Teilung zwischen Smartphones und Tablet-PCs statt, welche laut Google mit der nächsten Version wieder aufgehoben werden soll⁹.

Die Anzahl der verkauften und aktivierten Geräte nimmt täglich zu. Seit der Veröffentlichung von Android wurden weltweit mehr als 74,6 Millionen Smartphones mit Android ausgeliefert, wovon alleine im Jahr 2010 mehr als 67 Millionen verkauft wurden¹⁰. Android hatte dabei einen Marktanteil von 22,7 % aller weltweit verkauften Smartphones. Bei den benutzten Geräten „dürften weltweit noch Symbian und RIM in Führung liegen“¹¹.

Für Entwickler nicht ganz uninteressant ist die Aufteilung der einzelnen Android-Versionen: Die Version 2.2 (Froyo) hat einen Anteil von 63,9 %, Version 2.1 (Eclair) kommt auf 27,2 %. Die Versionen 1.5 und 1.6 kommen zusammen auf 6,2 % Anteil, während die neuesten Versionen 2.3 (Gingerbread) und 3.0 (Honeycomb) auf einen Anteil von 2,7 % kommen¹².

1.2 Grundlagen für Android

1.2.1 Benötigte Kenntnisse

Zur Entwicklung eigener Apps stehen an erster Stelle Kenntnisse in der Programmiersprache Java, denn „Java ist die Sprache der Wahl für die Android-Entwicklung“¹³. Als Entwickler kann man „den vollen Sprachumfang von Java 5“¹⁴ mit erweiterten Klassen für die Android-Entwicklung verwenden. Außerdem sind verschiedene Bibliotheken nutzbar, die z.B. zur Grafikdarstellung dienen. Der vom Compiler erzeugte Bytecode wird dann in der Dalvik Virtual Machine ausgeführt, einer speziell für mobile Geräte entwickelten virtuellen Maschine, kurz VM. Da diese VM sehr klein ist, wird jeder Prozess in einer eigenen VM ausgeführt¹⁵.

Jede App besteht aus mindestens einer Activity, einer erzeugten Oberfläche, mit der der Benutzer mit dem Gerät interagieren kann¹⁶. Da oft Speicherplatz be-

⁹vgl. [Mü 6]

¹⁰vgl.[Gar 6]

¹¹[Wir 6]

¹²vgl. [S 6b]

¹³[BLS10, S.90]

¹⁴[BLS10, S.90]

¹⁵vgl. [BP09, S.17f.]

¹⁶vgl. [BP09, S.20]

nötigt wird, hat jede Activity einen eigenen Lebenszyklus, der in der offiziellen Dokumentation genauer beschrieben wird¹⁷. Jede Activity besteht außerdem aus Views, die das Aussehen der Activity definieren¹⁸. Mithilfe von sogenannten Intents können Activities untereinander Daten austauschen¹⁹.

1.2.2 Benötigte Komponenten

„Vor dem ersten eigenen Android-Programm steht die Konfiguration einer geeigneten Entwicklungsumgebung“²⁰. Neben dem obligatorischen Android Software Development Kit (SDK) von Google muss man noch das Java Development Kit von Oracle installieren. Als Entwicklungsumgebung empfiehlt sich Eclipse, da Google dafür ein eigenes Plug-in anbietet²¹. Hilfe für die Installation findet man in der offiziellen Android-Dokumentation²². Wenn man alles installiert hat, kann man mit der Entwicklung beginnen.

1.3 Vorteile gegenüber anderen Systemen

Seit der Veröffentlichung konkurriert Android mit anderen Betriebssystemen für Smartphones, allen voran „iOS“ von Apple. Dabei wird Android „von Google klar als Konkurrent zum iPhone positioniert“²³. Im Gegensatz zum iPhone mit Apple als alleinigem Unterstützer hat Android mit der Open Handset Alliance ein großes Konsortium hinter dem System. Für Entwickler sind die Verwendung der etablierten Programmiersprache Java und die Möglichkeit der Entwicklung unter Windows, Mac OS X und Linux wichtig. Apple setzt für die Entwicklung von iPhone-Apps einen Mac voraus. Dank des offenen Quelltextes kann man bei Problemen bis in das Innerste debuggen. Aus Anwendersicht sprechen günstige Endgeräte und die Unterstützung von Flash für den Kauf eines Androidgerätes²⁴.

¹⁷<http://developer.android.com/guide/topics/fundamentals/activities.html>

¹⁸vgl. [BP09, S.38]

¹⁹vgl. [Lin10, S.189]

²⁰[Bec10, S.34]

²¹vgl. [Bec10, S.34]

²²<http://developer.android.com/sdk/index.html>

²³[Gla10, S.18]

²⁴vgl. [Gla10] und [BLS10, S.90f.]

2 Hauptteil

2.1 Die Idee zur App

Die Aufgabe des „Dortmund Guide“ soll darin bestehen, Touristen mehr Informationen über die Sehenswürdigkeiten Dortmunds mitzuteilen, damit diese mehr über die Stadt erfahren. Dabei sollen die Informationen selbst recherchiert und schnell abrufbar sein. Zwar gibt es im Android-Market schon ähnliche Applikationen, doch greifen diese entweder auf Wikipedia zurück²⁵, sind nur in englischer Sprache verfügbar²⁶ oder beschäftigen sich nur mit dem Fußballverein Borussia Dortmund. Auf Grundlage dieser Idee soll die App modelliert werden.

2.2 Die Modellierungsphase

Um eine möglichst kurze Ladezeit zu haben, soll beim Aufruf der App zunächst eine Startseite geöffnet werden. Die Startseite wird beim Betätigen des Buttons gelöscht und es soll dann eine neue Activity geöffnet werden, die dem Benutzer eine Auswahl an verschiedenen Sehenswürdigkeiten Dortmunds bietet. Mit dem Betätigen eines Buttons kommt man dann auf eine neue Seite, die die Sehenswürdigkeit mit einem Bild und mit Text vorstellt. Um Speicherplatz zu sparen, werden lediglich eine Klasse und eine Layout-Datei für die Sehenswürdigkeiten verwendet, die Texte und Bilder werden separat gespeichert und dynamisch in das Layout eingefügt. Bei Betätigung der Zurück-Taste des Handys in dieser Activity kommt man zurück zum Auswahlmenü.

2.3 Die Bestandteile

Nach der Ideenfindung und der Modellierung des „Dortmund Guide“ folgt nun die eigentliche Entwicklung. Diese erfolgt mit dem Android SDK und Eclipse 3.5 mit dem Android Plug-in. In diesem Kapitel werden die einzelnen Activities genauer beschrieben. Die kompletten Quelltexte befinden sich im Anhang.

Neben den hier vorgestellten Klassen und Dateien enthält das Projekt die Klasse

²⁵<https://market.android.com/details?id=com.envicon.stadtplan.Dortmund>

²⁶<https://market.android.com/details?id=com.fantasticapps.travelguidededortmund>

„R“, welche automatisch erzeugt und erweitert wird und deshalb hier nicht vorgestellt wird.

2.3.1 Die Startseite

Die Startseite dient dazu, den Benutzer beim Aufruf zu begrüßen. Die Startseite besteht aus der Klasse „startActivity“ und der Layout-Datei „main.xml“.

In main.xml wird die Anzeige eines Bildes definiert, welches sich im drawable-Ordner des Projektes befindet. Dafür wird ein ImageView zentriert im oberen Drittel des Bildschirmes verwendet. Unter dem ImageView befindet sich im unteren Drittel des Bildschirmes der Button, der zum Auswahlménü weiterleiten soll. Durch den Befehl

```
android:text="@string/start"
```

wird auf den String „start“ in „strings.xml“ verwiesen. Dank dieser Datei kann man alle Texte zentral verwalten und für verschiedene Sprachen einfach neue Strings hinzufügen und in der Layout-Datei auf diese verweisen. Durch die Beschreibungen in der Layout-Datei sieht die Startseite wie folgt aus:



Abbildung 1: Anzeige der Startseite

Die Klasse startActivity, welche die Klasse Activity erweitert, beinhaltet einen Button und den Befehl, eine neue Activity zu starten. Dieser Befehl wird beim Betätigen des Button ausgeführt. Dies wird mit

```
startActivity(new Intent(startActivity.this, menu.class));
```

erreicht. Wie man sieht, wird ein Intent erzeugt, der die Activity „menu“ öffnet. Wenn dies geschehen ist, wird diese Activity endgültig beendet.

2.3.2 Das Auswahlfenster

Das Auswahlfenster besteht aus der Klasse „menu“ und der Layout-Datei „menu.xml“. In dieser Activity kann der Benutzer wählen, über welche Sehenswürdigkeit er mehr erfahren möchte. Um diese Auswahl zu ermöglichen, werden in der Layout-Datei verschiedene Buttons mit derselben Höhe definiert. Auf Smartphones mit kleinen Bildschirmen kann es mit dieser Einstellung zu Schwierigkeiten beim Versuch, den letzten Button zu betätigen, kommen. Aus diesem Grund wird ein sogenannter ScrollView eingefügt. Er dient dazu, dass man bei kleinen Bildschirmen nach unten scrollen und Buttons betätigen kann, welche am Ende der Seite sind. Die Beschriftungen für diese Buttons sind auch in strings.xml gespeichert. Durch die Layout-Datei sieht das Auswahlfenster so aus:



Abbildung 2: Anzeige des Auswahlfensters

In der Klasse menu werden die Buttons initialisiert und es wird ein Intent deklariert. Außerdem wird definiert, was beim Klick eines Buttons geschehen soll. Dieser Befehl wird mit

```
intent = new Intent(menu.this, Sight.class);
```



```
intent.putExtra("clickedButton",1);
startActivity(intent);
onPause();
```

definiert. Zunächst erhält der Intent den Aufruf der Activity Sight. Danach wird in ihm der Wert 1 gespeichert. Der übergebene Wert ändert sich je nach gedrücktem Button, was in dieser Klasse durch eine if-Verzweigung, in der der betätigte Button abgefragt wird, erreicht wird. Zum Schluss wird dann die neue Activity aufgerufen und die aktuelle Activity wird pausiert.

2.3.3 Eine Beispiel-Sehenswürdigkeit

Um Speicherplatz zu sparen, werden für die Sehenswürdigkeiten eine gemeinsame Layout-Datei „sight.xml“ und eine gemeinsame Klasse Sight entwickelt. In Android wird diese Methode dadurch ermöglicht, dass man das Aussehen einer Activity dynamisch im Programmcode erzeugen kann.

Die hier verwendete Activity definiert in der Layout-Datei einen ImageView für ein Bild der Sehenswürdigkeit und zwei TextViews. Ein TextView dient für die Überschrift der Sehenswürdigkeit, der andere soll den Informationstext anzeigen. Im Programmcode der Klasse Sight wird mit

```
TextView t1 = (TextView)findViewById(R.id.TextView01);
t1.setTextSize(25);
```

die Schriftgröße des ersten TextViews, nach seiner vorherigen Initialisierung, auf 25 gesetzt. Analog dazu geschieht dasselbe für den zweiten TextView. Nachdem das geschehen ist, wird ein Index definiert, der den vom Intent gespeicherten Zahlenwert erhält. Dieser Vorgang wird durch

```
Intent intent = getIntent();
int index = intent.getIntExtra("clickedButton",0);
```

erreicht, wodurch eine Kopie des aus der vorherigen Activity aufgerufenen Intents erzeugt wird. Danach erhält die Indexvariable den gespeicherten Wert dieses Intents. Bei diesem Aufruf wird auch ein default-Wert angegeben, welcher in diesem Fall 0 beträgt.

Nach der Initialisierung des ImageView wird mithilfe von if-Verzweigungen das

Bild und der Text der jeweiligen Sehenswürdigkeit erzeugt. Dies geschieht dadurch, dass erstmal der Wert des Index abgefragt und dann das Bild und der Text eingefügt werden. Bei dem Beispiel der Reinoldikirche beschreibt folgender Code den Vorgang:

```
if(index == 1){  
    iv.setImageResource(R.drawable.reinoldikirche);  
    t1.setText(R.string.reinoldikirche);  
    t2.setText(R.string.rkirchetger);  
}
```

Der ImageView iv soll das Bild „reinoldikirche“ anzeigen. Der TextView t1 soll den String „reinoldikirche“ aus der Datei strings.xml anzeigen. Zum Schluss soll der TextView t2 den String „rkirchetger“ aus der Datei rkircheger.xml anzeigen. Das Ergebnis auf einem Endgerät sieht dann folgendermaßen aus:



Abbildung 3: Anzeige der Informationen zur Reinoldikirche

3 Fazit

3.1 Lernergebnisse

Mit der Darstellung der Entwicklung der hier vorgestellten App wurden erste Grundlagen über das Verständnis der App-Entwicklung für Android geschaffen. Für eine kleine Einführung wurden relativ viele, zum Teil auch schon fortgeschrittene, Themen behandelt. Mit diesen Kenntnissen können schnell brauchbare Ergebnisse erzielt werden. Außerdem wurde gezeigt, dass sich die Android-Plattform für die Entwicklung eines solchen Programmes eignet.

Die App wurde mithilfe des vom Android SDK bereitgestellten Emulators getestet, jedoch war die die Geschwindigkeit aufgrund der Emulation eines gesamten Endgerätes inklusive Hardware und Betriebssystem nicht zufriedenstellend. Deshalb wurde die Anwendung zusätzlich auf einem Endgerät getestet. Dort war die Anwendung schnell und konnte intuitiv bedient werden. Die Applikation sollte auf jeden Fall weiterentwickelt werden, um sie später für Android-Anwender anbieten zu können.

3.2 Mögliche Erweiterungen der App

Die vorgestellte App hat momentan noch nicht die Qualität, um mit ähnlichen Applikationen verglichen zu werden. Dafür fehlen ihr einige Eigenschaften. Dazu gehört z.B. die Ortung via GPS, wodurch sich die Möglichkeit ergibt, dass Sehenswürdigkeiten in der Nähe angezeigt werden können und man direkt zu ihnen navigiert werden kann. Außerdem hilft diese App in dieser Form nur weiter, wenn man den Namen der Sehenswürdigkeit kennt. Um dieses Problem zu lösen, kann die App mit Augmented Reality-Eigenschaften ausgestattet werden, sodass man als Anwender mithilfe der Kamera seines Smartphones die Sehenswürdigkeit erfassen kann und dann Informationen zu ihr erhält, entweder in einem neuen Fenster oder direkt im Bild. Um als vollwertiger Fremdenführer angesehen zu werden, sollten noch Informationen über Restaurants, Bars und ähnlichem abgerufen werden können. Diese Informationen können dann auch mit sozialen Netzwerken wie Facebook oder speziellen Bewertungsforen verknüpft werden. Mit all diesen zusätzlichen Eigenschaften und einer ansprechenden Gestaltung

kann die App dann ein Konkurrent zu schon erhältlichen Programmen sein und im Android Market angeboten werden.

3.3 Mögliche Entwicklung von Android

Der riesige Absatz gerade im Jahr 2010 und der dadurch entstandene Durchbruch von Android lässt auf eine unbeschwerte Zukunft für Entwickler hoffen. Das Marktforschungsunternehmen IDC sagt Android für 2011 einen Marktanteil von knapp 40% voraus, wodurch Android Marktführer bei den mobilen Betriebssystemen werden soll. Bis 2015 soll der Anteil sogar auf 45% steigen²⁷. Die potenzielle Zielgruppe für Android-Entwickler wächst auch dadurch, dass „auf dem BlackBerry Playbook, einem 7“ Tablet von RIM, [...] auch Android-Apps laufen“ werden²⁸. Dieser positive Eindruck wird allerdings durch neueste Meldungen getrübt.

So „plant Google, die eigene Kontrolle über das eigene Betriebssystem Android zu erhöhen“, indem Hersteller aus dem Early Access-Programm ausgeschlossen werden, wenn sie „nicht nach Google’s Pfeife tanzen“²⁹. Durch dieses Programm erhalten Hersteller relativ früh Einblick in eine neue Android-Version und können früher Handys mit einer neuen Version veröffentlichen. Außerdem hält Google den Quellcode der für Tablet-PCs optimierten Version 3.0 (Honeycomb) zurück, da es zurzeit sehr eingeschränkt auf Smartphones laufen soll³⁰. Aufgrund der momentanen Situation, dass man im Gegensatz zu iPhone-Apps als Entwickler wenig bei Android verdienen kann, entstehen neben dem offiziellen Android Market auch Markets vom Online-Kaufhaus amazon und dem Hardwarehersteller HTC, wodurch diese eine stärkere Kundenbindung erreichen wollen³¹. Durch diese Maßnahmen kann Android seine Offenheit und so die Unterstützer der ersten Stunde verlieren.

Die weitere Entwicklung Androids sowie anderer Smartphone-Betriebssysteme sollte weiter beobachtet und vor allem auch als Entwickler mitgestaltet werden und Gegenstand weiterer Untersuchungen sein.

²⁷vgl. [Bon 6]

²⁸[Dun 6]

²⁹[Br]

³⁰vgl. [S 6a]

³¹vgl. [Rö 6]

4 Anhang

4.1 Statistiken

Tabelle 1

Weltweite Smartphone-Verkäufe nach Betriebssystem in 2009 (in Tausend Einheiten)

Betriebssystem	Einheiten 2009	Marktanteil 2009 in %	Einheiten 2008	Marktanteil 2008 in %
Symbian	80.878,6	46,9	72.933,5	52,4
RIM	34.346,6	19,9	23.149,0	16,6
iPhone OS	24.889,8	14,4	11.417,5	8,2
Microsoft	15.027,6	8,7	16.498,1	11,8
Linux	8.126,5	4,7	10.662,4	7,6
Android	6.798,4	3,9	640,5	0,5
WebOS	1.193,2	0,7	0,0	0,0
Andere	1.112,4	0,6	4.026,9	2,9
Gesamt	172.373,1	100,0	139.287,9	100,0

Quelle: <http://www.gartner.com/it/page.jsp?id=1306513>

Tabelle 2

Weltweite Smartphone-Verkäufe nach Betriebssystem in 2010 (in Tausend Einheiten)

Betriebssystem	Einheiten 2010	Marktanteil 2010 in %	Einheiten 2009	Marktanteil 2009 in %
Symbian	111.576,7	37,6	80.878,3	46,9
Android	67.224,5	22,7	6.798,4	3,9
RIM	47.451,6	16,0	34.346,6	19,9
iOS	46.598,3	15,7	24.889,7	14,4
Microsoft	12.378,2	4,2	15.031,6	8,7
Andere	11.417,4	3,8	10.432,1	6,1
Gesamt	296.646,6	100,0	172.376,1	100,0

Quelle: <http://www.gartner.com/it/page.jsp?id=1543014>

4.2 Grafiken

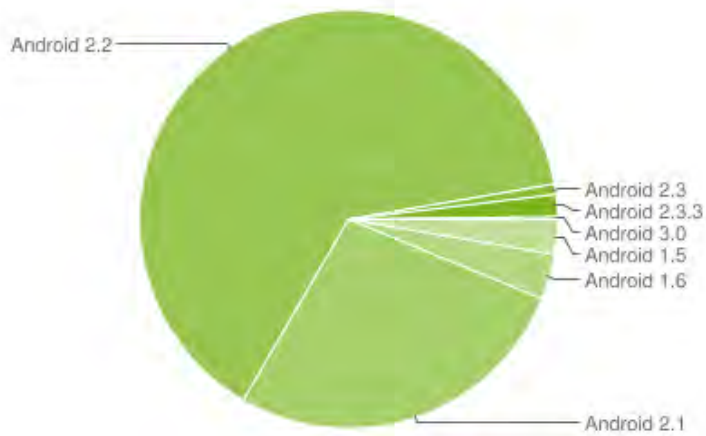


Abbildung 4: Die verschiedenen Versionen im Kreisdiagramm, Quelle: Google

Android Platform Versions

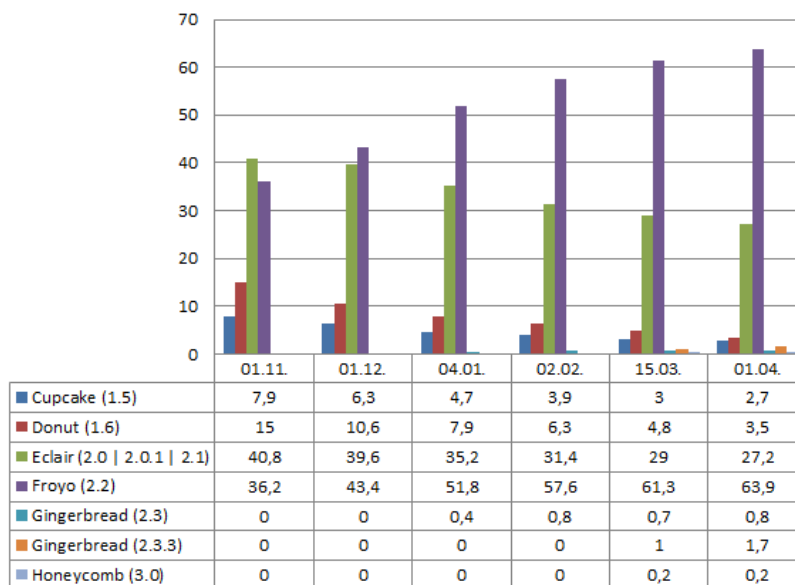


Abbildung 5: Die verschiedenen Versionen im Säulendiagramm, Quelle: Google

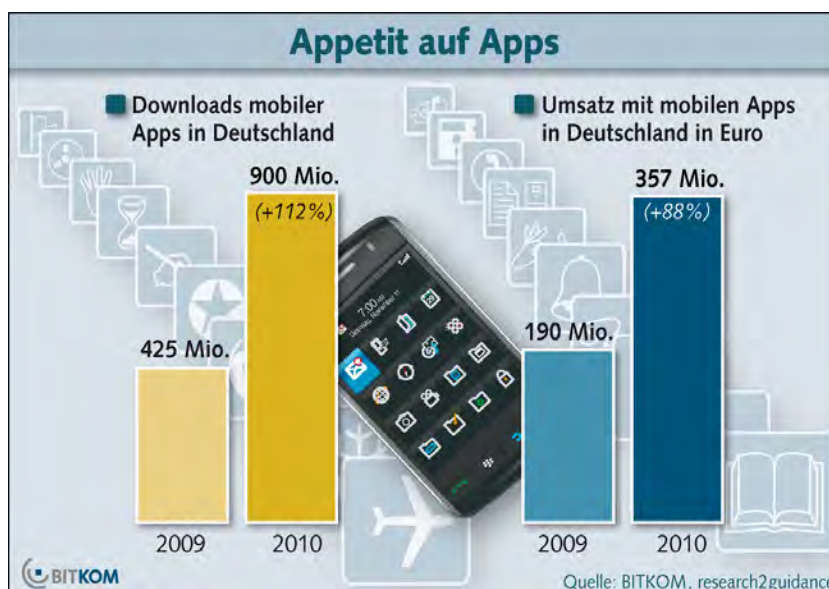


Abbildung 6: Umsatz mit App-Verkäufen in Deutschland, Quelle: Bitkom

4.3 Quelltexte

4.3.1 startActivity.java

```
package de.Dortmund.Guide;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;

public class startActivity extends Activity implements OnClickListener {
    private Button startButton;

    @Override
        public void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.main);

            startButton = (Button)findViewById(R.id.start);
            startButton.setOnClickListener(this);
        }

    @Override
        public void onClick(View v) {
            if(v == startButton){
                startActivity(new Intent(startActivity.this, menu.class));
                finish(); }
        }
    }
```

4.3.2 main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
<ImageView
    android:id="@+id/ImageView01"
    android:src="@drawable/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="50dip"
    android:layout_gravity="center">
</ImageView>
<Button
    android:id="@+id/start"
    android:layout_width="fill_parent"
    android:text="@string/start"
    android:layout_height="100dip"
    android:layout_marginTop="100dip"
    android:layout_marginLeft="50dip"
    android:layout_marginRight="50dip">
</Button>
</LinearLayout>
```

4.3.3 menu.java

```
package de.Dortmund.Guide;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
```



```
import android.view.View.OnClickListener;
import android.widget.Button;

public class menu extends Activity implements OnClickListener {
    private Button button1;
    private Button button2;
    private Button button3;
    private Button button4;
    private Button button5;

    @Override
        public void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.menu);

            //Initialisierung der Buttons
            button1 = (Button)findViewById(R.id.rkirche);
            button1.setOnClickListener(this);

            button2 = (Button)findViewById(R.id.stadion);
            button2.setOnClickListener(this);

            button3 = (Button)findViewById(R.id.westfalenpark);
            button3.setOnClickListener(this);

            button4 = (Button)findViewById(R.id.westfalahallen);
            button4.setOnClickListener(this);

            button5 = (Button)findViewById(R.id.zoo);
            button5.setOnClickListener(this);
        }

    @Override
        public void onClick(View v) {
            Intent intent;
```

```
if(v == button1){
    intent = new Intent(menu.this, Sight.class);
    intent.putExtra("clickedButton",1);
    startActivity(intent);
    onPause();
}
else if(v == button2){
    intent = new Intent(menu.this, Sight.class);
    intent.putExtra("clickedButton", 2);
    startActivity(intent);
    onPause();
}
else if(v == button3){
    intent = new Intent(menu.this, Sight.class);
    intent.putExtra("clickedButton", 3);
    startActivity(intent);
    onPause();
}
else if(v == button4){
    intent = new Intent(menu.this, Sight.class);
    intent.putExtra("clickedButton", 4);
    startActivity(intent);
    onPause();
}
else if(v == button5){
    intent = new Intent(menu.this, Sight.class);
    intent.putExtra("clickedButton", 5);
    startActivity(intent);
    onPause(); }
}
}
```

4.3.4 menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
    >
<ScrollView
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
    >
<Button
    android:id="@+id/rkirche"
    android:layout_width="fill_parent"
    android:text="@string/reinoldikirche"
    android:layout_height="100dip">
</Button>
<Button
    android:id="@+id/stadion"
    android:layout_width="fill_parent"
    android:text="@string/stadion"
    android:layout_height="100dip">
</Button>

<Button
    android:id="@+id/westfalenpark"
    android:layout_width="fill_parent"
    android:text="@string/westfalenpark"
    android:layout_height="100dip">
```

```
</Button>
<Button
    android:id="@+id/westfalahallen"
    android:layout_width="fill_parent"
    android:text="@string/westfalahallen"
    android:layout_height="100dip">
</Button>
<Button
    android:id="@+id/zoo"
    android:layout_width="fill_parent"
    android:text="@string/zoo"
    android:layout_height="100dip">
</Button>
</LinearLayout>
</ScrollView>
</LinearLayout>
```

4.3.5 Sight.java

```
package de.Dortmund.Guide;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.widget.ImageView;
import android.widget.TextView;

public class Sight extends Activity {

    @Override
    public void onCreate(Bundle savedInstanceState){
        super.onCreate(savedInstanceState);
        setContentView(R.layout.sight);
    }
}
```

```
TextView t1 = (TextView)findViewById(R.id.TextView01);
t1.setTextSize(25);

TextView t2 = (TextView)findViewById(R.id.TextView02);
t2.setTextSize(15);

Intent intent = getIntent();
int index = intent.getIntExtra("clickedButton",0);

ImageView iv = (ImageView)findViewById(R.id.ImageView01);

if(index == 1){
iv.setImageResource(R.drawable.reinoldikirche);
t1.setText(R.string.reinoldikirche);
t2.setText(R.string.rkirchetger);
}
else if(index == 2){
iv.setImageResource(R.drawable.apple);
t1.setText(R.string.stadion);
t2.setText(R.string.stadiontger);
}
else if(index == 3){
iv.setImageResource(R.drawable.apple);
t1.setText(R.string.westfalenpark);
t2.setText(R.string.wparktger);
}
else if(index == 4){
iv.setImageResource(R.drawable.apple);
t1.setText(R.string.westfalahallen);
t2.setText(R.string.whallentger);
}
else if(index == 5){
iv.setImageResource(R.drawable.apple);
t1.setText(R.string.zoo);
```

```
t2.setText(R.string.zootger); }  
}  
}
```

4.3.6 sight.xml

```
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    android:orientation="vertical"  
    android:layout_width="fill_parent"  
    android:layout_height="fill_parent">  
    <ImageView  
        android:id="@+id/ImageView01"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:layout_gravity="center">  
    </ImageView>  
    <TextView  
        android:id="@+id/TextView01"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:textStyle="bold">  
    </TextView>  
    <TextView  
        android:id="@+id/TextView02"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:textStyle="normal">  
    </TextView>  
</LinearLayout>
```

4.3.7 strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
```

```

<resources>
    <string name="app_name">Dortmund Guide</string>
    <string name="start">Start</string>
    <string name="reinoldikirche">Reinoldikirche</string>
    <string name="stadion">Signal-Iduna-Park</string>
    <string name="westfalenpark">Westfalenpark</string>
    <string name="westfalahallen">Westfalenhallen</string>
    <string name="zoo">Dortmund Tierpark</string>
</resources>

```

4.3.8 AndroidManifest.xml

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="de.Dortmund.Guide"
    android:versionCode="1"
    android:versionName="1.0">
    <application android:icon="@drawable/icon"
android:label="@string/app_name">
        <activity android:name=".startActivity"
            android:label="@string/app_name"
android:screenOrientation="portrait">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name="menu">
</activity>
<activity android:name="Sight"></activity>
</application>
    <uses-sdk android:minSdkVersion="5" />
</manifest>

```

Literatur

- [Bec10] BECKER, Arno: Bitte einsteigen: Die ersten Schritte zur eigenen App.
In: *Mobile Developer Android* 4/2010 (2010), S. 34–40
- [BLS10] BEIER, Andreas ; LINKE, Andreas ; SCHULZ, Hajo: Die eigene App:
Entwickeln für Android, iPhone, WebOS, Symbian, Blackberry und Win-
dows Mobile. In: *c't magazin für computertechnik* 16/2010 (2010), S.
90–95
- [Bon 6] BONSET, Sébastien: *Windows Phone 7 und Android laut Prognose
2015 vor iOS*. [http://t3n.de/news/
windows-phone-7-android-laut-prognose-2015-ios-303830/](http://t3n.de/news/windows-phone-7-android-laut-prognose-2015-ios-303830/).
Version: Rev. 2011-4-6
- [BP09] BECKER, Arno ; PANT, Markus: *Android: Grundlagen und Programmie-
rung*. Heidelberg : dpunkt.verlag GmbH, 2009
- [Br] BRÜGGEMANN, Christian: *Google will die Kontrolle über Android stär-
ken*. [http://www.androidpit.de/de/android/blog/394656/
Google-will-die-Kontrolle-ueber-Android-staerken](http://www.androidpit.de/de/android/blog/394656/Google-will-die-Kontrolle-ueber-Android-staerken). Version: Rev.
2011-4-6
- [Dir 6] DIRSCHERL, Hans-Christian: *Smartphone-Grundlagen: Android wird
Realität*. [http://www.pcwelt.de/ratgeber/
Android-wird-Realitaet-Smartphone-Grundlagen-1005340.html](http://www.pcwelt.de/ratgeber/Android-wird-Realitaet-Smartphone-Grundlagen-1005340.html).
Version: Rev. 2011-4-6
- [Dun 6] DUNSTHEIMER, Michael: *Android-Apps auf dem Blackberry Playbook
jetzt offiziell*. [http://www.androidpit.de/de/android/blog/394592/
Android-Apps-auf-dem-Blackberry-Playbook-jetzt-offiziell](http://www.androidpit.de/de/android/blog/394592/Android-Apps-auf-dem-Blackberry-Playbook-jetzt-offiziell).
Version: Rev. 2011-4-6
- [Gar 6] GARTNER: *Gartner Says Worldwide Mobile Device Sales to End
Users Reached 1.6 Billion Units in 2010; Smartphone Sales Grew 72
Percent in 2010*. <http://www.gartner.com/it/page.jsp?id=1543014>.
Version: Rev. 2011-4-6

- [Gla10] GLAHN, Kay: Newcomer mit Pfiff: Shootingstar Android. In: *Mobile Developer Android 4/2010* (2010), S. 18–20
- [Kö 6] KÖNIG, Kai: *(Fast) alles rund um Android: Von diesem Stern.* <http://www.heise.de/ix/artikel/Von-diesem-Stern-1134643.html>.
Version: Rev. 2011-4-6
- [Lin10] LINKE, Andreas: Appétitif: Einführung in die Entwicklung von Android-Apps, Teil 1. In: *c't magazin für computertechnik 22/2010* (2010), S. 188–191
- [Mü 6] MÜSSIG, Florian: *Android: Gemeinsamer Nachfolger für Gingerbread und Honeycomb.* <http://www.heise.de/newsticker/meldung/Android-Gemeinsamer-Nachfolger-fuer-Gingerbread-und-Honeycomb-1190944.html>. Version: Rev. 2011-4-6
- [Ris 6] RISSKA, Volker: *Downloads und Umsatz mit Apps steigen massiv.* <http://www.computerbase.de/news/consumer-electronics/kommunikation/smartphones/2011/februar/downloads-und-umsatz-mit-apps-steigen-massiv/>. Version: Rev. 2011-4-6
- [Rö 6] RÖHLINGER, Fabien: *„amazon(en) im Paradies?“ - Oder warum der amazon App Store im „walled garden“ einen Apfel gepflückt hat.* <http://www.androidpit.de/de/android/blog/394659/amazon-en-im-Paradies-Oder-warum-der-amazon-App-Store-im-walled-garden-einen-Apfel-gepflueckt-hat>. Version: Rev. 2011-4-6
- [S 6a] S, Anton: *Honeycomb Sourcecode wird zurückgehalten - Open Source Ära in Gefahr?* <http://www.androidpit.de/de/android/blog/394607/Honeycomb-Sourcecode-wird-zurueckgehalten-Open-Source-Aera-in-Gefahr>. Version: Rev. 2011-4-6
- [S 6b] S, Mario: *Android Platform Version - Zahlen für 01. April 2011.* <http://www.androidpit.de/de/android/blog/394681/Android-Platform-Version-Zahlen-fuer-01-April-2011>.
Version: Rev. 2011-4-6

[Wir 6] WIRTGEN, Jörg: *Smartphones: Android überholt Symbian, Apple verliert Marktanteile*. <http://www.heise.de/newsticker/meldung/Smartphones-Android-ueberholt-Symbian-Apple-verliert-Marktanteile-1180547.html>. Version: Rev. 2011-4-6

5 Eidesstattliche Erklärung

Ich erkläre, dass ich diese Facharbeit ohne fremde Hilfe angefertigt und nur die im Literaturverzeichnis angeführten Quellen benutzt habe.

Dortmund, 7.4.2011

Unterschrift des Schülers