

Institut für Parallele und Verteilte Systeme

Universität Stuttgart
Universitätsstraße 38
D-70569 Stuttgart

Fachstudie Nr. 172

Von Donuts, Pfefferkuchen und Sandwich-Eis

Paul Metzger, Ehssan Doust, Julia Holzschuh

Studiengang:	Softwaretechnik
Prüfer/in:	Prof. Dr.-Inf habil. Bernhard Mitschang
Betreuer/in:	Dipl.-Inf. Christoph Stach
Beginn am:	2012-12-03
Beendet am:	2013-06-03
CR-Nummer:	D.2.2

Kurzfassung

Am Beginn der Entwicklung einer Anwendung (App) für Android steht die Entscheidung, welche Plattform-Versionen unterstützt und welche Features angeboten werden. Das Ziel dieser Fachstudie ist es, bei dieser Entscheidung zu unterstützen. Hierzu werden als Erstes die Marktverteilung, sowie wichtige Teile des Android-Betriebssystems, des Android-SDKs und des Java-Frameworks beleuchtet. Anschließend werden nicht nur wichtige Abwägungen aufgezeigt und diskutiert, sondern auch relevante Features der Android-Plattform herausgearbeitet, kategorisiert und bewertet. Die Bewertung berücksichtigt hierbei auch die Entwicklung der Features, um einen direkten Vergleich der Plattform-Versionen zu ermöglichen. Dadurch versetzt dieses Dokument den Leser in die Lage, eine ausgewogene und begründete Entscheidung für eine bestimmte Plattform-Version zu treffen.

Inhaltsverzeichnis

1. Einleitung	9
1.1. Aufbau des Dokuments	9
1.2. Motivation	9
1.3. Android	9
1.3.1. Versionen	10
1.3.2. Google SDK	10
2. Vorstellung der API Versionen	11
2.1. Überblick	11
2.2. Grundfunktionen	12
2.3. Donut	13
2.3.1. Plattform-Version 1.6	13
2.4. Eclair	14
2.4.1. Plattform-Version 2.0	14
2.4.2. Plattform-Version 2.1	15
2.5. Gingerbread	16
2.5.1. Plattform-Version 2.3	16
2.6. Honeycomb	17
2.6.1. Plattform-Version 3.0	17
2.6.2. Plattform-Version 3.1	18
2.6.3. Plattform-Version 3.2	18
2.7. Ice Cream Sandwich	19
2.7.1. Plattform-Version 4.0	19
2.8. Jelly Bean	20
2.8.1. Plattform-Version 4.1	20
2.9. Gegenüberstellung der Features	21
2.10. Reflections	26
2.11. Ziel- und Mindest-API	27
2.12. Beispiel App Farbmischer	29
2.12.1. Motivation	29
2.12.2. Beschreibung	30
2.12.3. Fazit	30
3. Bewertung der Plattform-Versionen	33
3.1. Allgemeiner Eindruck	46
4. Fazit und Ausblick	49

A. Anhang	51
Literaturverzeichnis	63

Abbildungsverzeichnis

2.1. Android-Architektur	12
2.2. Verteilung von Plattform-Versionen am 03.01.2013	28
2.3. Festlegung der Plattform-Versionen mit dem SDK	29
2.4. Farbmischer	30

Tabellenverzeichnis

2.1. App Components Überblick	21
2.2. Barrierefreiheit Überblick	21
2.3. Benutzerfreundlichkeit Überblick	22
2.4. Eingabegeräte Überblick	22
2.5. Enterprise Überblick	22
2.6. Hardware Überblick	23
2.7. Kamera Überblick	23
2.8. Konnektivität Überblick	24
2.9. Multimedia Überblick	25
2.10. Sensoren Überblick	25
2.11. Textbearbeitung Überblick	26
2.12. User Interface Überblick	26
3.1. App Komponenten Bewertung	34
3.2. Barrierefreiheit Bewertung	34
3.3. Benutzerfreundlichkeit Bewertung	35
3.4. Eingabegeräte Bewertung	35
3.5. Enterprise Bewertung	36
3.6. Hardware Bewertung	37
3.7. Kamera Bewertung	38
3.8. Konnektivität Bewertung	40
3.9. Multimedia Bewertung	41
3.10. Sensoren Bewertung	42
3.11. Textbearbeitung Bewertung	43

3.12. User Interface Bewertung	45
--	----

Verzeichnis der Listings

A.1. MainActivity.java	52
A.2. MainActivity.java	53
A.3. MainActivity.java	54
A.4. ColorHelper.java	55
A.5. ColorHelper.java	56
A.6. ColorClickListener.java	56
A.7. ColorDragListener.java	57
A.8. ColorDragListener.java	58
A.9. ColorTouchListener.java	59
A.10. ColorTouchListener.java	60
A.11. activity_main.xml	61
A.12. activity_main.xml	62

1. Einleitung

Es folgt eine Einleitung über den Aufbau des Dokuments, die Motivation und Hintergrundinformationen dieser Arbeit. Dieses Dokument adressiert Personen, die

- genaue Informationen zu den Funktionalitäten einer Plattform-Version benötigen.
- entscheiden, welche Plattform-Version eine Anwendung (App) unterstützen soll.

1.1. Aufbau des Dokuments

Das zweite Kapitel beschreibt, zu Beginn, die Grundfunktionen des Android-Betriebssystem und stellt danach detailliert die untersuchten Plattform-Versionen vor. Das Kapitel schließt mit jeweils einem Unterkapitel zu Reflections sowie zu Ziel- und Mindest-API. Das dritte Kapitel stellt die Entwicklung der einzelnen Features, über die Plattform-Versionen hinweg, dar. Das Vierte Kapitel schließt die Fachstudie mit einem Fazit ab und gibt einen Ausblick auf zukünftige Plattform-Versionen.

1.2. Motivation

Am Beginn der Entwicklung einer Anwendung (App) für Android steht die Entscheidung, welche Plattform-Versionen unterstützt und welche Features angeboten werden. Bei dieser Entscheidung spielen mehrere Faktoren eine Rolle. Wichtig sind die von der zu entwickelnden App benötigten Features, aber in vielen Fällen auch die Verteilung der Plattform-Versionen innerhalb der Zielgruppe, letzteres aufgrund mangelnder Abwärtskompatibilität der Apps. Im folgenden Dokument wird anstelle von der Verteilung innerhalb der Zielgruppe von der Marktverteilung gesprochen. Dieses Dokument versetzt den Leser in die Lage, eine ausgewogene und begründete Entscheidung zugunsten einer bestimmte Plattform-Version zu treffen.

1.3. Android

Das Android-Betriebssystem wurde in erster Linie für mobile Geräte wie Smartphones und Tablets entwickelt. Es wird jedoch auch auf anderen Geräten wie zum Beispiel Fernsehern, Netbooks, DVD-Playern und Kameras eingesetzt. Das Betriebssystem nutzt einen

1. Einleitung

angepassten Linux-Kernel, Apps werden in der ebenfalls angepassten Java Virtual Machine - Dalvik - ausgeführt. Die Entwicklung von Apps wird von Google in Java und C/C++ unterstützt. Entwickelte Apps können über den Google Play Store oder manuell von einem Speichermedium oder per Android Debug Bridge (ADB) installiert werden. Die ADB ist ein Kommandozeilen-Werkzeug, das viele Funktionalitäten bietet, die einen Entwickler bei der Entwicklung und insbesondere beim Debuggen einer App unterstützen. Die ADB kann unter anderem per USB genutzt werden.

1.3.1. Versionen

Die untersuchten Plattform-Versionen sind:

- Version 1.6, Donut
- Version 2.0, Eclair
- Version 2.1, Eclair
- Version 2.3, Gingerbread
- Version 3.0, Honeycomb
- Version 3.1, Honeycomb
- Version 3.2, Honeycomb
- Version 4.0, Ice Cream Sandwich
- Version 4.1, Jelly Bean

1.3.2. Google SDK

Das Google SDK ist teil der Android Developer Tools (ADT). Teil des SDKs ist ein Eclipse-Plugin, das den Entwickler bei der Entwicklung, dem Test und dem Deployment unterstützt. Es beinhaltet nicht nur die nötigen Bibliotheken, um gegen die einzelnen Plattform-Versionen zu entwickeln, sondern auch Werkzeuge. Werkzeuge sind beispielsweise ein Emulator, ein Dalvik Debug Monitor Server (DDMS), ein Draw 9-Patch, ein Layoutopt und ein Zipalign. DDMS bietet viele Funktionen um Informationen eines Android Gerätes auszulesen oder das Gerät zu manipulieren. So können unter anderem Informationen über Threads und den Speicher ausgelesen werden, aber auch Anrufe und SMS simuliert werden. Layoutopt unterstützt bei der Optimierung des Layouts und der Layout-Hierarchie. Zipalign optimiert APK-Files, die zur Installation von Apps genutzt werden, sodass die ausgeführte App weniger RAM belegt.

2. Vorstellung der API Versionen

2.1. Überblick

In diesem Kapitel werden einige Android Application Programming Interfaces (API) und die beinhalteten Features vorgestellt. Nach dem "Lexikon der Informatik" von Fischer und Hofer [FH08] ist eine API eine:

"Anwendungsprogrammier-Schnittstelle, welche die Menge aller zulässigen Systemaufrufe der „anprogrammierten“ Instanz (Protokollschicht, Gerät, Controller usw.) beschreibt; die API gibt detailliert an, wie ein Dienstaufruf an diese Instanz zu programmieren und zu verwenden ist."

Der Begriff Feature spielt im folgenden eine zentrale Rolle, daher wird hier seine Bedeutung in dieser Arbeit erläutert. Ein Feature ist Teil einer API und stellt dem Programmierer eine Funktionalität zur Verfügung. Beispiel für ein Feature ist die Möglichkeit Daten eines Hörsensors auszulesen. Die Lesbarkeit der durch die API angebotenen Methoden ist beispielsweise kein Feature.

Die erste Android-API wurde im Oktober 2008 herausgebracht. Wie bei vielen Softwareprodukten wurde auch Android im Laufe der Jahre verbessert. Die Änderungen wurden wie im Softwarebereich üblich mit Versionsnummern versehen. Bei signifikanten Änderungen wurde die Hauptversion um eins hochgezählt, kleine Änderungen oder Bugfixes wurde lediglich die Nebenversion hochgezählt. Änderungen im API-Framework wurden zusätzlich mit Plattform-Levels versehen, diese wurden in fortlaufender Zahl vergeben, sodass sie anhand dieser Nummer genau identifizierbar sind. So kann es sein das verschiedenen Versionsnummern dasselbe API-Level haben, solange sich bei den Änderungen der verschiedenen Versionen keine Änderungen in der API ergaben. Zum Beispiel besitzen die Versionen 2.3, 2.3.1 und 2.3.2 alle das API-Level 9. Die API-Levels bestimmen die Geräte, auf welchen die App ausgeführt werden kann.

Die Wahl des API-Level sollte vom Entwickler möglichst sorgfältig durchdacht werden, da hier konträre Interessen eine Rolle spielen. Näheres hierzu finden sie in Kapitel 2.11. Die Plattform-Versionen 3.0 und 3.1 sind nur für Tablet-PCs.

In dem folgenden Kapitel 2.2 wird ein Überblick über die vorhandenen Grundfunktionen vor der Version 1.6 gegeben. Danach werden die Versionen 1.6, 2.0, 2.1, 3.0, 3.1, 3.2, 4.0 und 4.1 vorgestellt. In Kapitel 2.9 gibt es eine Auflistung verschiedener Features und welche Versionen diese Features beinhalten. Danach folgt das Kapitel 2.10 über Reflections

in Android. Im nächsten Kapitel wird die in Android vorhandene Ziel- und Mindest-API genauer erklärt. Im letzten Kapitel 2.12 wird ein Beispiel für den Unterschied der einzelnen Plattform-Versionen aufgezeigt. Als Quelle für diese Kapitel diente die offizielle Android Homepage für Entwickler [And12].

2.2. Grundfunktionen

In diesem Abschnitt werden die Grundfunktionen der Android API beschrieben. Die folgenden Abschnitte bauen auf diesem auf. Insbesondere beinhaltet die in dem jeweiligen Abschnitt beschriebene Version auch die Features der Vorversionen und die Grundfunktionen.

Abbildung 2.1 veranschaulicht den Zusammenhang der verschiedenen Android-Komponenten. Auf unterster Ebene steht der Linux Kernel, welcher die Dalvik Runtime und diverse System Libraries ausführt. Die Dalvik Runtime und die System Libraries werden von den systemweit zur Verfügung stehenden Frameworks verwendet, z.B. dem Notification Manager. Alle Android Apps - unabhängig davon, ob es sich um mit ausgelieferte Apps oder nachträglich installierte Apps handelt - sitzen auf oberster Ebene, und können die System-Frameworks und Funktionalitäten der anderen Android Apps verwenden.

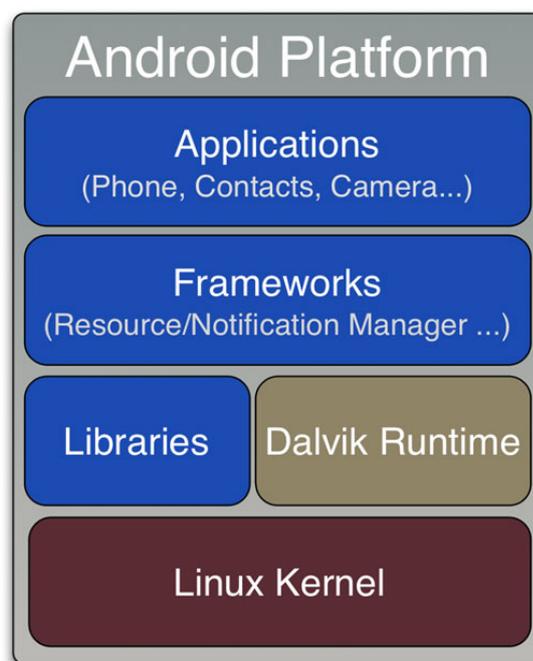


Abbildung 2.1.: Android-Architektur - <http://apcmag.com/whats-android.htm>

Die ersten Versionen des Android-Betriebssystems waren primär für Smartphones ausgelegt. Daher bieten die ersten Plattform-Versionen alle Grundfunktionen der damaligen Handys. Es gibt die Möglichkeit Daten in der, in Android integrierten, SQLite Datenbank abzulegen. Weiter werden Bibliotheken bereit gestellt um grafikbezogene Berechnungen durchzuführen. Um mit anderen Geräten kommunizieren zu können, kann WiFi, Bluetooth oder GPRS in Verbindung mit verschiedenen Netzwerktechnologien genutzt werden. Insbesondere gibt es einige Klassen um HTTP-Verbindungen aufzubauen und zu verwalten. Auch sind die ersten Plattform-Versionen dafür geeignet mit Multimediadaten umzugehen.

Da über die Android API die Funktionalität anderer Programme genutzt werden kann, werden im Folgenden Apps aufgeführt, die standardmäßig bei einer Installation von Android mitgeliefert werden.

Dazu gehört eine Anwendung zum versenden, empfangen und verwalten von SMS, eine App die Kontakte verwaltet, ein E-Mail-Client, eine App zum abspielen von Musik im mp3-Format, ein Webbrowser, die Möglichkeit Bilder aufzunehmen, die Möglichkeit GPS zu nutzen, eine Wecker-App und eine App um andere Personen anzurufen (Telefon-App). Android bietet aber auch seit der ersten Version viele Möglichkeiten um das Smartphone zu personalisieren. Beispielsweise kann ein eigener Klingelton oder ein Bildschirmhintergrund gewählt werden. Außerdem gehört der Google Play Store, ehemals Android Market, zur Grundausstattung von Android-Smartphones. Der Google Play Store bietet Apps, Bücher und Multimediaangebote zum freien oder kostenpflichtigen Download an.

Ebenfalls zu Android gehört die speziell für mobile Geräte optimierte Java Runtime Environment (JRE) Dalvik. Es ist vorgesehen, dass Apps in Dalvik ausgeführt werden, da Dalvik unter anderem weniger Akku und Speicher verbraucht als die Standard JRE. Android bietet aber auch die Möglichkeit in C und C++ geschriebene Apps nativ auszuführen.

2.3. Donut

2.3.1. Plattform-Version 1.6

Die Plattform-Version 1.6 ist seit September 2009 auf dem Markt verfügbar. Hierbei ergaben sich vor allem einige Verbesserungen für die Entwickler als auch im API Framework. Zu den wichtigsten Änderungen gehört unter anderem die Unterstützung verschieden großer Bildschirme und VPN. Des Weiteren wurden einige Elemente für den Entwickler freigegeben.

Aktualisierte Features

In dieser Version wurde die Kamera aktualisiert. Es besteht nun die Möglichkeit der Mehrfachauswahl bei Fotos. Der Wechsel zwischen der Kamera und der Video Funktion erfolgt

2. Vorstellung der API Versionen

schneller und die allgemeine Performance wurde verbessert. Des Weiteren wurde die Open-Core2 media engine erweitert und unterstützt nun den OpenMAY Encoder, zusätzliche Audio Formate in der AuthorEngine sowie einen verteilten Puffer im Decoder. Das neue XML Attribut android:onClick bietet eine Schnittstellen Definition aus einer Layout Datei an, um einen Callback zu implementieren. Die Suche auf dem Gerät wurde durch eine Quick Search Box erweitert und durch das benutzerdefinierte einstellen des Suchradius optimiert. Darüber hinaus wurde der Linux Kernel auf die Version 2.6.29 upgedatet.

Neue Features

Mit dieser Version ergaben sich einige Neuerungen bezüglich der App und deren Entwicklung. In der App besteht nun die Möglichkeit den Standort zu ermitteln, nach expliziter Erlaubnis auf einen externen Speicher zu schreiben und auf die Browser Historie mittels Read oder Write zu zugreifen. Des Weiteren ist es Möglich die maximale und vorhergesehene API für die App anzugeben und deren minimalen Anforderungen an das Gerät. Mit dieser Version bringt Android auch ein Framework zur Unterstützung der Barrierefreiheit mit, sodass es dem Entwickler nun möglich ist Plugins für die Barrierefreiheit zu erstellen. Eine weitere Neuerung ist die Unterstützung von Virtual Private Network. Für das Energiemanagement gibt es jetzt eine neue Ansicht mit der sich die Programme und deren Verbrauch überwachen lassen. Ein neues Gesten Framework bietet die Möglichkeit Gesten zu erstellen, zu laden und zu speichern und sie mit Aktionen zu verknüpfen. Außerdem bietet Android mit dieser Version Text-to-speech an, mit verschiedenen Sprachen und deren Akzent. Im Telefonie-Stack wird nun auch CDMA für US-Mobilfunknetze unterstützt und die Mehrfachverbindung von Wi-Fi ist möglich. Zur allgemeinen Verbesserung des Benutzererlebnisses wurde eine Schnellstartleiste für Schnittstellen wie WLAN und Bluetooth hinzugefügt. Außerdem werden mit dieser Version verschiedene Bildschirmauflösungen unterstützt.

2.4. Eclair

2.4.1. Plattform-Version 2.0

Die Plattform-Version 2.0 ist seit Ende Oktober 2009 auf dem Markt. Im Dezember 2009 gab es dann noch ein Bugfix-Update. Der Schwerpunkt dieser Version liegt hauptsächlich auf der Bedienbarkeit für den Durchschnitts-User, dadurch ergaben sich wenige Neuerungen aber umso mehr Aktualisierungen.

Aktualisierte Features

Mit dieser Version wurden wurde einige neue Funktionen für die Kamera hinzugefügt. Hierfür gibt es nun eine eingebaute Blitzlicht Unterstützung, einen digitalen Zoom, einen Szene Modus, einen Weißabgleich, einen Farbeffekt und einen Makrofokus. Des Weiteren gab

es ein Update auf Bluetooth 2.1, es gibt noch eine Ein- und Ausschaltfunktion, die Möglichkeit sich via RFCOMM zu verbinden sowie die Geräte und Service Erkennung. Außerdem wurde die Architektur im graphischen Bereich für die Verbesserung der allgemeine Performance geändert und es gibt eine automatische Generierung von Vorschaubildern bei neu erstellten Images. In der Accountverwaltung besteht nun die Möglichkeit mehrere Accounts zu verwalten und zu synchronisieren. Es gibt die neue Funktion "Quick Contact" mit derer eine schnellere und individuelle Kontaktaufnahme möglich ist. Darüber hinaus hat der Entwickler nun die Möglichkeit Synchronisationsadapter zu erstellen, um Kontakte aus zusätzlichen Datenquellen leichter zu synchronisieren. Außerdem gibt es eine Suchfunktion speziell für alle gespeicherten SMS und MMS und die Möglichkeit ein Limit für eine Speicherung einer Konversation festzulegen. Dadurch wird beim Erreichen des Limits die ältesten Nachrichten automatisch gelöscht. Die virtuelle Tastatur hat mit dieser Version ein verbessertes Layout bekommen und Unterstützung nun auch Multi-touch. Das Wörterbuch speichert automatisch neue Wörter und fügt die Namen der Kontakte hinzu. Ab dieser Version unterstützt der Browser nun auch teilweise HTML5 und den Zoom bei Doppelklick. Es gibt einen direkten Zugriff auf die Browser-Bar und die Lesezeichen besitzen nun ein Vorschaubild. Im Kalender wird nun auch der Teilnahmestatus der Eingeladenen Personen angezeigt und in der Agenda Ansicht wird das endlos Scrollen unterstützt. Diese Version bringt auch die Unterstützung der mehrfach Berührung bis zu 3 Punkten mit sich. Außerdem werden KeyEvents unterstützt, wie zum Beispiel langes berühren, action-on-up und abbrechen eines Klick-Vorganges.

Neue Features

Neuerungen ergab es vor allem bei den Apps. Hier wird nun eine Unterstützung im Umgang mit dem Lebenskreis der Apps angeboten. Der Entwickler hat nun auch die Möglichkeit den Bildschirm zum Leben zu erwecken auch wenn das Gerät noch gesperrt ist. Des Weiteren besteht die Möglichkeit den Dockingstatus zu übertragen. Darüber hinaus wurde ein zentraler Account Manager eingeführt mit dessen Hilfe man zum Beispiel die Benutzernamen und Passwörter verwalten kann.

2.4.2. Plattform-Version 2.1

Die Plattform-Version 2.1 ist seit Januar 2010 auf dem Markt. Es gibt wenig bedeutendes gegenüber der vorherigen Plattform Android 2.0. Der Schwerpunkt der Version 2.1 liegt vor allem auf der Speicheroptimierung.

Aktualisierte Features

Mit dieser Version werden Callbacks bereitgestellt, welche Änderungen des Verbindungsstatus bereitstellen. Außerdem werden Informationen über das aktuell benutzte Netzwerk Signal bereitgestellt. Des Weiteren ergaben sich Änderungen bezüglich der Ansicht und des WebKits. In der Ansicht gibt es nun die Möglichkeit abzufragen ob Scrollbars vorhanden

2. Vorstellung der API Versionen

sind und ob die App undurchlässig ist. Darüber hinaus besteht die Möglichkeit alle Views zu einer Ansicht zu löschen oder neue Views als Kinder hinzuzufügen und deren Reihenfolge abzufragen oder festzulegen. Im WebKit gibt es einige neue Methoden um den Speicherbereich zu manipulieren, den Standort zu ermitteln und ihn im Browser anzuzeigen, den Cache sowie den Speicher und den Zoom zu verwalten und für den Umgang mit Videos, Browser Historie, Grenzen des Caches und mehr.

Neue Features

Mit dieser Version neu hinzugekommen sind, in den App Components, die animierten Wallpapers für die Apps.

2.5. Gingerbread

2.5.1. Plattform-Version 2.3

Die Plattform-Version 2.3 wurde im Oktober 2010 veröffentlicht, der Schwerpunkt dieser Version liegt vor allem auf der verbesserten Benutzerschnittstelle.

Aktualisierte Features

Zu den Aktualisierungen dieser Version zählt vor allem die Überarbeitung des UI Framework, dieses enthält nun viele Verbesserungen und Erweiterungen. Außerdem wurden neue Content Provider hinzugefügt.

Neue Features

Einige Features sind in dieser Version neu hinzugekommen - hierzu zählen unter anderem neue APIs, welche die Erstellung von VoIP-Apps unterstützen. Apps können nun NFC-Tags lesen. Es sind neue Sensoren hinzugekommen, welche von Android nun unterstützt werden, u.a. das Gyroskop, der Accelerometer, ein Gravitationsensor sowie ein Barometer. Alle Kameras eines Android-Geräts können nun von Apps gezielt angesteuert werden. Ein globaler Download-Manager wurde eingeführt. Außerdem wurde im Rahmen der Verbesserungen am UI Framework auch eine Unterstützung für besonders große Bildschirme hinzugefügt.

2.6. Honeycomb

2.6.1. Plattform-Version 3.0

Die Plattform-Version 3.0 wurde am 23. Februar 2011 erstmalig veröffentlicht und beinhaltet vor allem Weiterentwicklungen, welche auf eine bessere Integration von Tablets hinzielen. Die Schwerpunkte dieser Plattform-Version liegen in den Bereichen Benutzerschnittstelle (und deren einfache Skalierung auf unterschiedliche Bildschirmgrößen), sowie den Bereichen 2D- und 3D-Animationen und Konnektivität. Außerdem wurden diverse Enterprise-Features erweitert und aktualisiert.

Aktualisierte Features

Die Zwischenablage kann nun mehr als textuellen Inhalt aufnehmen, so können in ihr unter anderem auch URIs und Intents abgelegt werden. Für die Benutzerschnittstelle sind nun mehrere neue Widget-Typen verfügbar, und die Statusbarbenachrichtigungen wurden erweitert, sodass nicht mehr nur rein textuelle Benachrichtigungen möglich sind. Im Bereich Konnektivität wurden die Bluetooth APIs erweitert, nun können innerhalb von Apps angeschlossene Headsets erkannt werden. Das WebKit wurde erweitert, und der Browser wurde aktualisiert. Dieser ermöglicht nun unter anderem die Darstellung von 3D-Effekten, welche in CSS 3 definiert wurden. Die DRM APIs wurden erweitert, und ermöglichen nun das Hinzufügen eigener DRM-Schemata.

Neue Features

Im Bereich der Benutzerschnittstelle ersetzt die ActionBar nun die herkömmliche Titelleiste und ermöglicht so das einfache Bereitstellen wichtiger Befehle. Das Kontextmenü kann ab sofort komplett durch die ActionBar ersetzt werden. Sogenannte Fragments ermöglichen die einfache Modularisierung von Activities, indem sie Teile der Benutzerschnittstelle und Aktionen kapseln. Das neue Animations-Framework ermöglicht die Animation beliebiger Eigenschaften von UI-Elementen. Mit den neuen Transformationen ist es möglich, einfache 2D- und 3D-Animationen auf beliebige UI-Elemente anzuwenden. Die Hardware-Beschleunigung für 2D-Elemente kann nun global im Anwendungsmanifest aktiviert werden. Mehrere Views können nun simultan Touch-Events empfangen. Das "Holografische Theme" wird auf alle Anwendungen, welche mindestens für Android 3.0 geschrieben werden, angewendet. Die Umsetzung von Drag and Drop wird nun durch vordefinierte APIs vereinfacht. Zudem sind einige neue Widget-Typen hinzugekommen, unter anderem ein Kalenderwidget und ein Suchwidget. Desktop-Tastaturen werden nun in vollem Umfang unterstützt.

2.6.2. Plattform-Version 3.1

Die Plattform-Version 3.1 wurde Mai 2011 veröffentlicht. Die Schwerpunkte dieser Version liegen in den Bereichen USB-Unterstützung, Benutzerschnittstelle und Browser.

Aktualisierte Features

Im Bereich der Benutzerschnittstelle wurde eine Reihe neuer Eingabegeräte und Touch-Events hinzugefügt. Die Größe von App Widgets können nun geändert werden. Außerdem wurden das Animations-Framework und das UI-Framework erweitert. Der Download-Manager wurde erweitert und bietet nun unter anderem die Möglichkeit, auf abgeschlossene Downloads zu reagieren. Die für Entwickler offiziell zugänglichen Netzwerk-APIs wurden um das RTP (Real-Time Transport Protocol) erweitert. Das WebKit wurde insgesamt verbessert und erweitert, außerdem unterstützt der Browser nun das Abspielen von HTML5-Videos, welche mit dem Video-Tag angegeben werden.

Neue Features

Über USB lassen sich nun eine große Bandbreite an Geräten anschließen und verwenden. Außerdem ist für die Kamera die sogenannten MTP/PTP API hinzugekommen.

2.6.3. Plattform-Version 3.2

Die Plattform-Version 3.2 wurde erstmalig am 17.07.2011 veröffentlicht, der Schwerpunkt dieser Version liegt hauptsächlich in der Anpassung an unterschiedliche Bildschirmgrößen und -auflösungen.

Aktualisierte Features

Die MTP API wurde aktualisiert, sodass Apps nun direkt mit Dateien, welche auf angeschlossenen SD-Karten liegen, interagieren können. Außerdem können Apps Mediadateien direkt von einer angeschlossenen SD-Karte lesen. Das UI-Framework wurde erweitert.

Neue Features

Im Bereich der Benutzerschnittstelle gibt die neue Screens Support API dem Entwickler eine starke Kontrolle darüber, wie eine App auf unterschiedlichen Bildschirmgrößen angezeigt wird. Im Bereich Hardware wurde die USB-Unterstützung erweitert, sodass nun die USB-Deskriptoren von angeschlossenen Geräten abgefragt werden können.

2.7. Ice Cream Sandwich

2.7.1. Plattform-Version 4.0

Mit dieser Plattform-Version - welche am 19.10.2011 erschienen ist - sind viele Features, die Android 3.0 Tablets vorbehalten waren, nun auf Smartphones verfügbar. Mit dieser Plattform-Version wurde die Benutzeroberfläche stark verbessert und es werden mehr Möglichkeiten geboten um mit anderen Geräten zu kommunizieren.

Aktualisierte Features

Mit dieser Version wurde der Umgang mit Multimediadaten weiter verbessert, so ist es nun möglich den Multimediaplayer über eine andere App zu steuern. Außerdem werden mehr Formate zum abspielen von Multimedia Daten unterstützt. Passend dazu sind ,mit Android 3.x in den Browser integrierte, Multimedia Funktionen nun auch für Smartphones verfügbar. Aber auch der Umgang mit Sensor Daten wurde verbessert, so können nun zum einen Luftfeuchtigkeits- und Temperatursensoren ausgelesen werden zum anderen wurden die Funktionen zu Nutzung der Gesichtserkennung und des Autofokus erweitert. Mögliche Funktionalitäten die eine App anbieten kann wurden erweitert, so können Apps nun Einträge im Kalender, Voicemails und Rechtschreibprüfungen anlegen aber auch einen VPN-Dienst anbieten. Die Nutzeroberfläche bietet nun mehr Möglichkeiten Sehbehinderten Barrierefreiheit zu bieten. Dies ist aber nicht die einzige Verbesserung der Nutzeroberfläche. Popupmenüs können ab dieser Version per XML beschrieben werden, außerdem kann geprüft werden ob das Menü geschlossen wurde. Kinder eines Gridlayout können selbst bestimmen wie sie angeordnet werden. Die ActionBar, welche die für Tablets eingeführte Systembar ersetzt, passt sich nun besser an Bildschirmgrößen an und bietet mehr Element. Des weiteren ist die Grafikbeschleunigung nun für alle Fenster per Standard aktiviert. Grafikeffekte können durch die GPU berechnet und durch OpenGL direkt dargestellt werden. Auch Sicherheitsaspekte der API wurden verbessert so ist es nun möglich den Zugriff auf die Kamera zu verbieten und per API auf den Zertifikat-Schlüsselbund zuzugreifen. Zu Letzt ist zu erwähnen das die JNI erweitert wurde um durch Pointer verursachte Fehler zu finden.

Neue Features

Diese Version der Plattform bringt viele Neuerung in Bezug auf Konnektivität. So können nun Daten aus sozialen Netzwerken nur anhand der Kontakte gefunden werde. Es wurde ein Invite Intent eingeführt um Kontakte aus Apps heraus, zu sozialen Netzwerken hinzuzufügen. Weiter wurde die auf Near Field Communication (NFC) basierende Technologie Android Beam eingeführt die zum Datenaustausch zwischen Geräten dient. Für den Datenaustausch wurde aber auch Wi-Fi Direct eingeführt, das eine Punkt-zu-Punkt Verbindung zwischen Geräten ermöglicht. Medizinische Geräte können ab dieser Version mit den Geräte per Bluetooth verbunden werden, die Netzwerkauslastung so wie Statistiken

2. Vorstellung der API Versionen

über welchen Medien der Besitzer Kontakte erreicht können abgefragt werden. Außerdem ist es nun möglich Profile für den Gerätebesitzer zu erstellen. Für den Umgang mit Multimedia Inhalten wurde ein TextureView eingeführt der es ermöglicht Videos anzuzeigen. Für die Nutzeroberfläche wurde ein Switch Widget eingeführt das sich durch Schieben und Berühren verändern lässt. Es gibt nun ein Klasse um komfortabel zwischen Einstellungen hin und her schalten zu können. Außerdem ist nun kein Hardwareknopf nötig um das Optionsmenü aufzurufen.

2.8. Jelly Bean

2.8.1. Plattform-Version 4.1

Die Plattform-Version 4.1, welche am 27.06.2012 erschienen ist, stellt einige Aktualisierungen für ein besseres Look and Feel bereit. Es wurde eine Schnittstelle geschaffen, um externe Wiedergabegeräte anzuschließen. Außerdem bietet Google nun einen Dienst an um kurze Nachrichten auf Geräte zu schicken, ähnlich dem Apple Push Notification Service (APNS). Außerdem ist einer der Schwerpunkte eine neue Technologie mit der es möglich ist eine flüssige Bedienung zu gewährleisten.

Aktualisierte Features

Mit dieser Version wurde die Barrierefreiheit um viele Funktionen erweitert. So bieten, unter anderem, mit Standardviews erstellte Apps Barrierefreiheit an. Apps wurden aber nicht nur um Barrierefreiheit erweitert, sondern auch um die Möglichkeit den Aufruf von Activities mit Animationen zu begleiten. Benachrichtigungen in der ActionBar können nun große Bilder und viele Buttons erhalten, außerdem können sie mit verschiedenen Styles und Prioritäten versehen werden. Eine weitere Verbesserung der Nutzeroberfläche ist, dass das GridLayout und ViewStub in Widgets genutzt werden kann. Das mit Android 4.0 eingeführte Android Beam bietet nun auch die Möglichkeit Bluetooth zu nutzen um höhere Transferraten zu erreichen. Per Wi-Fi kann ab dieser Version aktiv nach Diensten in der Umgebung gesucht werden. Zu Letzt ist zu erwähnen, dass auch der Umgang mit Multimediadaten verbessert wurde, so können nun low-level Media Codecs abgefragt werden, Renderscript Computation kann genutzt werden und die Webview ist performanter.

Neue Features

Zu den neuen Features, bezüglich der Nutzeroberfläche, zählt ein 16 Millisekunden VSync Heartbeat durch den eine flüssige Darstellung garantiert wird. Außerdem kann nun Triple Buffering für ein besseres Rendering genutzt werden. Ab dieser Version ist es für Apps möglich den Systemzustand aus dem Kernel, mittels Systrace zu erhalten. Außerdem können

Apps benachrichtigt werden falls neue Eingabegeräte angeschlossen werden und die Funktionen bereits angeschlossener auslesen. Solche externen Geräte wurden um die Funktion zu vibrieren erweitert. Eine weitere Neuerung ist das Google Cloud Messaging for Android (GCM) mit dem, ähnlich wie mit Apples Push Notification Service (APNS), Kommandos und Nachrichten an Geräte und Apps geschickt werden können. Ab dieser Version wird eine Schnittstelle für Audio via USB angeboten, außerdem kann Mehrkanalton per HDMI ausgegeben werden. Auch ist es möglich Audioaufnahmen noch während der Aufnahme zu bearbeiten und zu wählen auf welchen Geräten die Wiedergabe stattfinden soll.

2.9. Gegenüberstellung der Features

Der Aufbau und die Einteilung der folgenden Tabellen richtet sich nach der Dokumentation der offiziellen Android-Homepage. [And12]

App Components	Version								
	1.6	2.0	2.1	2.3	3.0	3.1	3.2	4.0	4.1
Memory management	x	x	x	x	x	x	x	x	x
Content providers	x	x	x	x	x	x	x	x	x
Live Wallpapers			x	x	x	x	x	x	x
App Backup				x	x	x	x	x	x
Isolated services									x
benutzerdefinierte App Stack Navigation									x

Tabelle 2.1.: App Components Überblick

Barrierefreiheit	Plattform-Version								
	1.6	2.0	2.1	2.3	3.0	3.1	3.2	4.0	4.1
Gesprochenes Feedback	x	x	x	x	x	x	x	x	x
Haptisches Feedback	x	x	x	x	x	x	x	x	x
Töne bei App Aktivitäten									x

Tabelle 2.2.: Barrierefreiheit Überblick

2. Vorstellung der API Versionen

Benutzerfreundlichkeit	Plattform-Version								
	1.6	2.0	2.1	2.3	3.0	3.1	3.2	4.0	4.1
Text to Speech	x	x	x	x	x	x	x	x	x
Wischgesten	x	x	x	x	x	x	x	x	x
Speech to Text			x	x	x	x	x	x	x
Download Manager			x	x	x	x	x	x	x
Drag and Drop					x	x	x	x	x
Telefonbesitzer als „Kontaktkarte“ (Telefonbesitzer hat Identität)								x	x

Tabelle 2.3.: Benutzerfreundlichkeit Überblick

Eingabegeräte	Plattform-Version								
	1.6	2.0	2.1	2.3	3.0	3.1	3.2	4.0	4.1
Downloadbare IME	x	x	x	x	x	x	x	x	x
Spiele-Controller						x	x	x	x
Tastatur / Maus						x	x	x	x
Vibrierende Eingabegeräte									x

Tabelle 2.4.: Eingabegeräte Überblick

Enterprise	Plattform-Version								
	1.6	2.0	2.1	2.3	3.0	3.1	3.2	4.0	4.1
Permissions	x	x	x	x	x	x	x	x	x
Virtual Private Network (VPN)	x	x	x	x	x	x	x	x	x
Mobile Device Management Policies				x	x	x	x	x	x
Digital Rights Management (DRM)					x	x	x	x	x
Zertifikatmanagementfunktionen								x	x

Tabelle 2.5.: Enterprise Überblick

2.9. Gegenüberstellung der Features

Hardware	Plattform-Version								
	1.6	2.0	2.1	2.3	3.0	3.1	3.2	4.0	4.1
Energieverwaltung	x	x	x	x	x	x	x	x	x
Smartphone	x	x	x	x				x	x
Speicherkarten	x	x	x	x	x	x	x	x	x
Apps können auf Speicherkarte installiert werden			x	x	x	x	x	x	x
Apps können auf externen Speicher installiert werden				x	x	x	x	x	x
Tablet					x	x	x	x	x
Apps können direkt von Speicherkarte lesen							x	x	x

Tabelle 2.6.: Hardware Überblick

Kamera	Plattform-Version								
	1.6	2.0	2.1	2.3	3.0	3.1	3.2	4.0	4.1
Autofokus	x	x	x	x	x	x	x	x	x
Fotografieren während Videoaufnahme								x	x
Gesichtserkennung	x	x	x	x	x	x	x	x	x
Rausch Unterdrückung bei Aufnahme									x

Tabelle 2.7.: Kamera Überblick

2. Vorstellung der API Versionen

Konnektivität	Plattform-Version								
	1.6	2.0	2.1	2.3	3.0	3.1	3.2	4.0	4.1
Global System for Mobile Communications (GSM)	x	x	x	x	x	x	x	x	x
General Packet Radio Service (GPRS)	x	x	x	x	x	x	x	x	x
Enhanced Data Rates for GSM Evolution (EDGE)	x	x	x	x	x	x	x	x	x
Code Division Multiple Access (CDMA)	x	x	x	x	x	x	x	x	x
Universal Mobile Telecommunications System (UMTS)	x	x	x	x	x	x	x	x	x
Wireless Local Area Network (WLAN)	x	x	x	x	x	x	x	x	x
Bluetooth		x	x	x	x	x	x	x	x
High Speed Packet Access (HSPA)		x	x	x	x	x	x	x	x
Near Field Communication (NFC)				x	x	x	x	x	x
Portable Hotspot				x	x	x	x	x	x
Long Term Evolution (LTE)					x	x	x	x	x
Push-to-talk						x	x	x	x
Realtime transport protocol (RTP)						x	x	x	x
Voice over IP (VOIP)						x	x	x	x
Android Beam								x	x
Netzwerkverbrauch								x	x
Wi-Fi Direct								x	x
Network service discovery									x

Tabelle 2.8.: Konnektivität Überblick

Multimedia	Plattform-Version								
	1.6	2.0	2.1	2.3	3.0	3.1	3.2	4.0	4.1
Aufnahme von Audio-Rohdaten	x	x	x	x	x	x	x	x	x
Media codecs	x	x	x	x	x	x	x	x	x
Musical Instrument Digital Interface (MIDI)	x	x	x	x	x	x	x	x	x
Wiedergabe von Audio-Rohdaten	x	x	x	x	x	x	x	x	x
Audioeffekte				x	x	x	x	x	x
HTTP Live Stream					x	x	x	x	x
Audioaufnahme ab Signal									x
Lückenlose Wiedergabe									x
Timed Text Tracks									x

Tabelle 2.9.: Multimedia Überblick

Sensoren	Plattform-Version								
	1.6	2.0	2.1	2.3	3.0	3.1	3.2	4.0	4.1
Annäherungssensor	x	x	x	x	x	x	x	x	x
Drucksensor	x	x	x	x	x	x	x	x	x
Global Positioning System (GPS)	x	x	x	x	x	x	x	x	x
Gyroskop	x	x	x	x	x	x	x	x	x
Helligkeit	x	x	x	x	x	x	x	x	x
Höhensensor	x	x	x	x	x	x	x	x	x
Kompass	x	x	x	x	x	x	x	x	x
Accelerometer				x	x	x	x	x	x
Gravitation				x	x	x	x	x	x
Feuchtigkeit								x	x
Temperatur								x	x

Tabelle 2.10.: Sensoren Überblick

2. Vorstellung der API Versionen

Textbearbeitung	Plattform-Version								
	1.6	2.0	2.1	2.3	3.0	3.1	3.2	4.0	4.1
Autovervollständigung	x	x	x	x	x	x	x	x	x
Copy and Paste	x	x	x	x	x	x	x	x	x
Markieren	x	x	x	x	x	x	x	x	x
Speech to Text			x	x	x	x	x	x	x
Copy and Paste komplexer Daten					x	x	x	x	x
Drag and Drop					x	x	x	x	x
Rechtschreibkorrektur					x	x	x	x	x

Tabelle 2.11.: Textbearbeitung Überblick

User Interface	Plattform-Version								
	1.6	2.0	2.1	2.3	3.0	3.1	3.2	4.0	4.1
Location Manager	x	x	x	x	x	x	x	x	x
Notifications	x	x	x	x	x	x	x	x	x
Text to Speech	x	x	x	x	x	x	x	x	x
Verschiedene Bildschirmauflösungen	x	x	x	x	x	x	x	x	x
Online-Konten Verwaltung		x	x	x	x	x	x	x	x
Quick Contact		x	x	x	x	x	x	x	x
Backup Service				x	x	x	x	x	x
Car Mode				x	x	x	x	x	x
Cloud to Device Messaging (C2DM)				x	x	x	x	x	x
Multitouch				x	x	x	x	x	x
Night Mode				x	x	x	x	x	x
Grafikbeschleunigung					x	x	x	x	x
Action Bar					x	x	x	x	x
Unterstützt 720p (Fernseher)							x	x	x

Tabelle 2.12.: User Interface Überblick

2.10. Reflections

Reflections sind eine Möglichkeit APIs zu nutzen die, um die Abwärtskompatibilität zu gewährleisten, nicht angeboten werden.

Ist beispielsweise eine App auf vielen Geräten einer frühen Version installiert möchte man einerseits nicht die Kompatibilität mit diesen Versionen verlieren. Andererseits möchte man

die neuen APIs auf Geräten einer neueren Version nutzen. Um dieses Dilemma zu lösen können mit Hilfe von Reflections Klassen höherer Plattform-Versionen instantiiert werden.

Dieses Vorgehen schränkt jedoch sowohl die Lesbarkeit als auch die Testbarkeit ein. Bevor eine Klasse instantiiert werden kann muss überprüft werden ob diese Klasse existiert. Erst danach kann man sicher sein, dass die Klasse ohne Fehler instantiiert werden kann. Die Überprüfung bringt sehr unübersichtlichen Code mit sich, da einerseits einige Fehler abgefangen werden müssen, andererseits ist dieses Vorgehen sehr unüblich und kann daher schwer zu verstehen sein. Ein weitere Nachteil entsteht dadurch, dass der Klassenpfad als String übergeben wird und so Eclipse nicht prüfen kann ob der Klassenpfad in mindestens einer Plattform-Version valide ist.

2.11. Ziel- und Mindest-API

Es gibt derzeit viele Android-Geräte auf dem Markt, welche nicht mit der aktuellsten Plattform-Version laufen. Viele von den älteren Geräten lassen sich nur bis zu einer bestimmten Plattform-Version upgraden. Der Grund hierfür ist, dass viele Android-Geräte nicht die Standard-Android-Distribution von Google verwenden können (z.B. wegen Hardware-Inkompatibilitäten), und das Verteilen von Updates vom Gerätehersteller vorgenommen wird. Hierbei werden ältere Smartphones häufig ignoriert, da für die Hersteller die Kosten, für die Anpassung der Plattform-Version an das Gerät, zu hoch sind. Dies führt zu einer starken Fragmentierung der Plattform-Versionen auf dem Markt.

Die Abbildung 2.2 zeigt die Verteilung der Plattform-Version auf dem Markt am 03.01.2013. Die Angaben der Besitzverhältnisse sind in der Einheit Prozent gehalten. Es wird deutlich, dass viele Android-Benutzer noch die Version Gingerbread (2.3) installiert haben, gefolgt von Ice Cream Sandwich (4.0). Um möglichst viele Geräte direkt und ohne Hilfsmittel nutzen zu können, muss der Entwickler einen Kompromiss, zwischen den neuesten Android-Features und einer möglichst großen Zielgruppe finden. Um zu gewährleisten das die App nur auf den vom Entwickler vorgesehenen Geräten installiert wird, kann im Android-Manifest die Ziel- und die Mindest-API festgelegt werden. Das Android-Manifest ist eine XML-Datei. Jede App benötigt ein solches Manifest. Der Name der Datei muss immer `AndroidManifest.xml` betragen und sie muss im Stammverzeichnis liegen. Das Manifest beinhaltet die wichtigsten Informationen über die App für die Plattform-Version. Ohne diese Informationen kann die App nicht gestartet werden. Unter anderem existiert in diesem Manifest das Element `<uses-sdk />`. Mit diesem Element lässt sich die Kompatibilität zu verschiedenen API-Levels definieren. Der Google Play Store liest diese Daten aus dem Anwendungsmanifest aus und zeigt dem Benutzer nur solche Apps an, die auch auf dem verwendeten Gerät laufen. Falls der User ein Gerät mit einem API-Level unter der vom Entwickler angegebenen Mindest-API besitzt, so erscheint bei der Installation ein Hinweis, dass die Plattform-Version des Gerätes kleiner als die mindest Version der App ist. Danach wird die Installation abgebrochen.

Die Plattform-Mindestversion wird mit dem Attribut `android:minSdkVersion` festgelegt.

2. Vorstellung der API Versionen

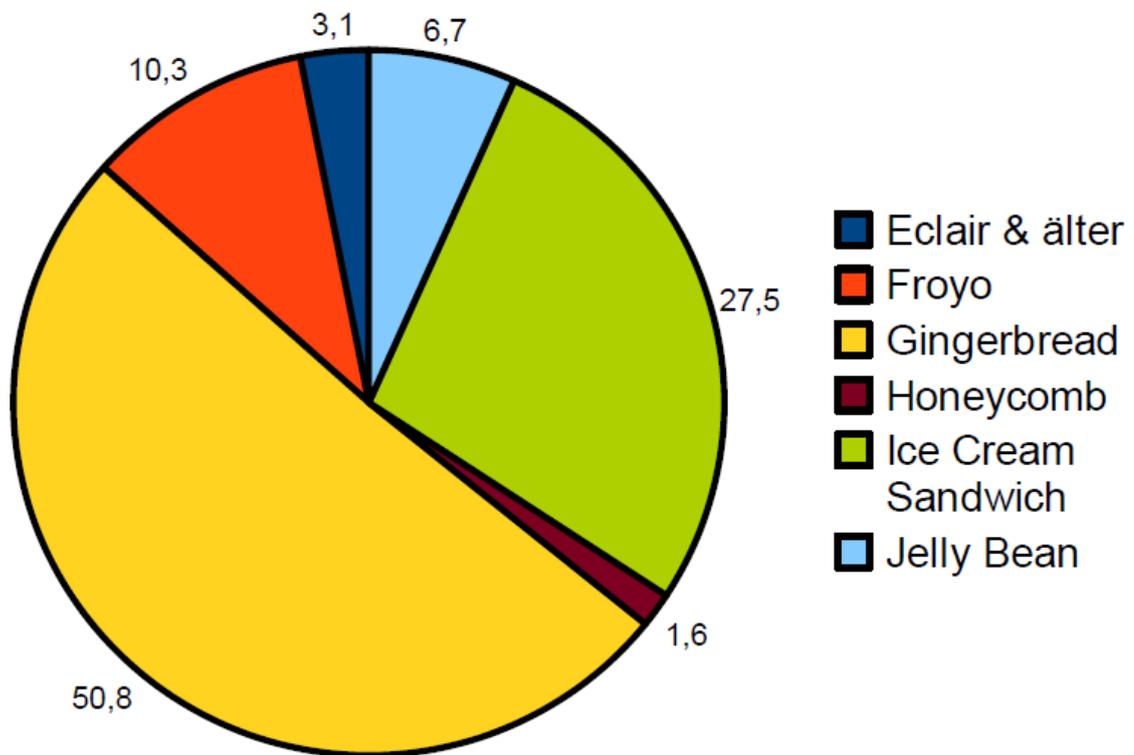


Abbildung 2.2.: Verteilung von Plattform-Versionen am 03.01.2013 - <http://developer.android.com/about/dashboards/index.html>

Hier wird das minimale API-Level der benötigten Version eingetragen. Android-Apps, welche hier ein höheres API-Level als das Gerät selbst besitzen, können nicht installiert werden. Wenn diese Attribut nicht definiert wird, so ist der Standardwert 1.

Die Plattform-Zielversion wird durch das Element `android:targetSdkVersion` definiert. Hier wird das optimale API-Level, welches vom Entwickler gründlich getestet wurde, angegeben. Wenn dieses Attribut nicht definiert wird, so entspricht es der Plattform-Mindestversion. Für dieses API-Level wurden bei der Entwicklung keinerlei Kompatibilitätsmodi vorgesehen. Wenn die Plattform-Geräteversion sich von der `android:targetSdkVersion` unterscheidet, so kann das erfolgreiche Ausführen der App auf zweierlei Arten realisiert werden. Entweder es wird die Plattform-Version durch den Aufruf `VERSION.SDK_INT` geprüft und die Methoden mittels Verzweigungen an die vorhandene Plattform-Version angepasst. Oder es werden klassischen Reflections, siehe Abschnitt 2.10, benutzt [AS12]. Abgesehen von der Ziel- und Mindest-API gibt es auch die Maximal-API, welche mit dem Attribut `android:maxSdkVersion` festgelegt werden kann. Diese verbietet die Installation von Apps auf Geräten mit einer höheren API. Dieses Attribut wird nur bis zur Plattform-Version 2.0.1 unterstützt.

In Eclipse lassen sich die Plattform-Versionen gleich beim Anlegen eines Projekts festlegen, siehe Abbildung 2.3. Die `AndroidManifest.xml` wird daraufhin automatisch generiert.

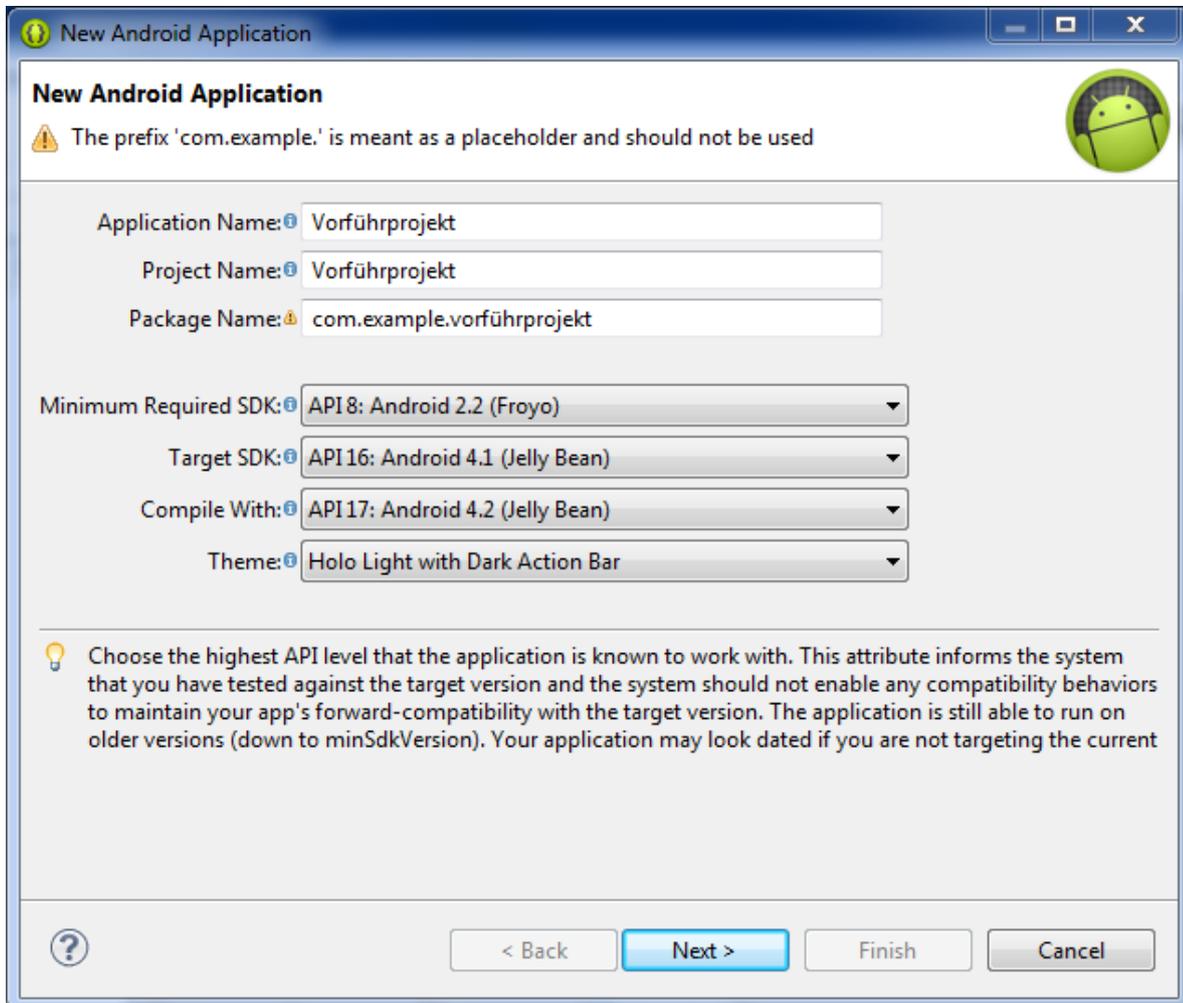


Abbildung 2.3.: Festlegung der Plattform-Versionen mit dem SDK

2.12. Beispiel App Farbmischer

2.12.1. Motivation

Die App “Farbmischer“ dient dazu den Unterschied zwischen Plattform-Versionen beispielhaft darzustellen.

2.12.2. Beschreibung

Die App bietet die Möglichkeit Farben aus 3 bestehenden Grundfarben zu mischen. Wird die App auf Plattform-Version 3.0 und höher ausgeführt, wird dafür ein Drag and Drop Listener genutzt. Der Nutzer zieht die Farben dann in ein großes Feld um sie dort zu mischen. Wird die App auf einer niedrigeren Plattform-Version ausgeführt, wird der Drag and Drop Listener durch den entsprechende Click-Listener ersetzt. In diesem Fall tippt der Nutzer auf die Farbe die er dem Mischfeld hinzufügen will.

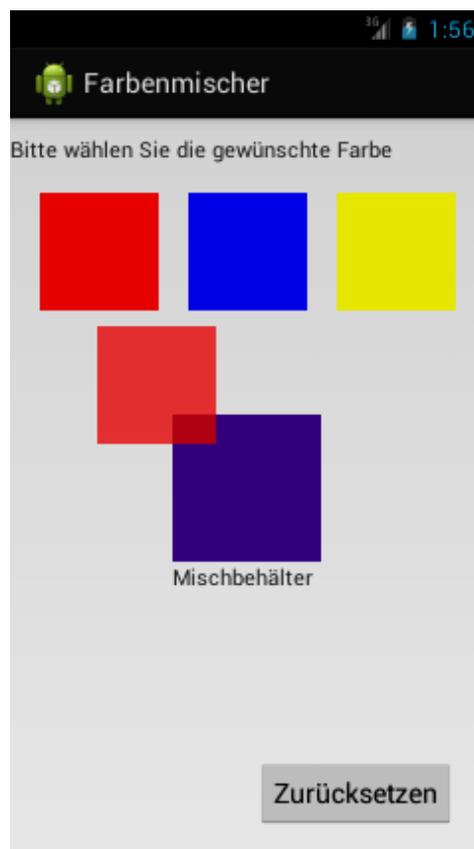


Abbildung 2.4.: Farbmischer

2.12.3. Fazit

Die App zeigt, dass durch das Nutzen neuer Features Apps an Usability gewinnen können. Wird die App ohne Drag and Drop Listener ausgeführt, büßt sie merkbar an Usability ein, da das Mischen von Farben per Drag and Drop viel mehr dem Mischen von realen Farben

ähnlich ist als das Mischen per antippen.
Der Quellcode der App befindet sich im Anhang.

3. Bewertung der Plattform-Versionen

Im folgenden Kapitel wird die Entwicklung der Plattform-Versionen mit Hilfe der folgenden Bewertungsstufen untersucht:

Einführung - "E"

Das Feature wird in der offiziellen Plattform-Version eingeführt.

Kleine Änderungen - "K"

Das Feature erfährt Bugfixes, kleine Änderungen und/oder ein Refactoring. Änderungen beziehen sich auf einzelne Methoden, Attribute oder Klassen, die keine zentrale Rolle spielen.

Abgrenzung: Es wurden keine neuen Sinneinheiten hinzugefügt oder komplett überarbeitet.

Aktualisierung - "A"

Das Feature wird stark aktualisiert, die Aktualisierungen können von Entwicklern sofort wahrgenommen werden. Große Refactorings, die Überarbeitung kompletter Sinneinheiten oder Änderungen an der Architektur werden ebenfalls unter diesem Punkt zusammengefasst.

Abgrenzung: Es wurden keine neuen Sinneinheiten hinzugefügt.

Neue Funktionen - "N"

Das Feature wird um neue Sinneinheiten erweitert, beispielsweise werden weitere Technologien unterstützt.

Abgrenzung: Es wurden kein Versionsprung vollzogen.

Versionssprung - "V"

Das Feature oder die damit verbundene Technologie erfährt einen Versionssprung. Die Hauptversion wird in der Plattform-Version erhöht.

Keine Änderungen - "-"

Es gibt keine oder keine offensichtlichen Änderungen in diesem Versionssprung der Plattform-Version.

Noch nicht vorhanden - " "

Das Feature ist in der offiziellen Plattform-Version noch nicht vorhanden.

Durch diese Bewertungsstufen soll sichergestellt werden, dass der Leser in möglichst kurzer Zeit einen optimalen Überblick, über die Menge und die Art von eingeführten Neuerungen erhält. Anschließend kann der Leser auf Grundlage dieser Informationen eine oder mehrere zu einer gewissen Problemstellung passende Plattform-Versionen ermitteln und sich daraufhin auf eine bestimmte Version festlegen.

App Komponenten	Version								
	1.6	2.0	2.1	2.3	3.0	3.1	3.2	4.0	4.1
Memory management	-	-	-	K	-	-	-	N	K
Content providers	K	K	-	-	A	-	-	K	K
Live Wallpapers				E	A	-	-	-	N
App Backup				E	-	-	-	A	-
Isolated services									E
benutzerdefinierte App Stack Navigation									E

Tabelle 3.1.: App Komponenten Bewertung

Barrierefreiheit	Plattform-Version								
	1.6	2.0	2.1	2.3	3.0	3.1	3.2	4.0	4.1
Gesprochenes Feedback	E	-	-	-	-	-	N	-	K
Haptisches Feedback	E	-	-	-	-	-	N	-	K
Töne bei App Aktivitäten									E

Tabelle 3.2.: Barrierefreiheit Bewertung

Benutzerfreundlichkeit	Plattform-Version								
	1.6	2.0	2.1	2.3	3.0	3.1	3.2	4.0	4.1
Text to Speech	E	-	-	K	K	-	-	A	-
								Multiple Text to Speech Engine	
Wischgesten	E	-	-	A	-	-	-	-	-
Speech to Text				K	-	-	-	K	-
				E(2.2)					
Download Manager				E	N	K	-	-	K
Drag and Drop					E	-	-	-	-
Telefonbesitzer als „Kontaktkarte“ (Telefonbesitzer hat Identität)								E	-

Tabelle 3.3.: Benutzerfreundlichkeit Bewertung

Eingabegeräte	Plattform-Version								
	1.6	2.0	2.1	2.3	3.0	3.1	3.2	4.0	4.1
(Downloadbare) IME	-	-	-	-	K	-	-	-	-
Spiele-Controller						E	-	-	-
Tastatur / Maus						E	-	-	-
Vibrierende Eingabegeräte									E

Tabelle 3.4.: Eingabegeräte Bewertung

Enterprise	Plattform-Version								
	1.6	2.0	2.1	2.3	3.0	3.1	3.2	4.0	4.1
Permissions	x	x	x	x	x	x	x	x	x
Virtual Private Network (VPN)	N	-	-	-	-	-	-	N	-
	UI L2TP/ PPTP							VPN- Service	
Mobile Device Management Policies				K	-	A	-	A	-
						Verschlüsselung / Pass- wort		Kamera	
Digital Rights Management (DRM)					E	-	-	-	-
Zertifikatmanagement- funktionen								E	K
									Ändere Trusted Storage

Tabelle 3.5.: Enterprise Bewertung

Hardware	Plattform-Version								
	1.6	2.0	2.1	2.3	3.0	3.1	3.2	4.0	4.1
Energieverwaltung	-	-	K	K	-	-	-	-	-
			Bildsch. an	Reboot					
Speicherkarten	-	-	-	N	-	-	N	-	K
			Apps auf SD (2.2)				Apps können direkt von Speicher lesen		Erlaubnis zum le- sen
Apps können auf Speicherkarte installiert werden			E	-	-	-	-	-	-
Apps können auf exter- nen Speicher installiert werden				E	-	-	-	-	-
Tablet					E	-	-	V	-
								Für Smart- phone und Tablet opti- miert	
Apps können direkt von Speicherkarte lesen							E	-	K
									Erlaubnis zum le- sen

Tabelle 3.6.: Hardware Bewertung

Kamera	Plattform-Version								
	1.6	2.0	2.1	2.3	3.0	3.1	3.2	4.0	4.1
Autofokus	-	-	-	-	-	-	-	K	K
								Continous Auto Fo- cus (CAF)	
Fotografieren während Videoaufnahme								E	-
Gesichtserkennung	E	-	-	-	-	-	-	N	-
								Teile des Gesichts erken- nen	
Rausch Unterdrückung bei Aufnahme									E

Tabelle 3.7.: Kamera Bewertung

Konnektivität	Plattform-Version								
	1.6	2.0	2.1	2.3	3.0	3.1	3.2	4.0	4.1
Global System for Mobile Communications (GSM)	-	-	-	-	-	-	-	-	-
General Packet Radio Service (GPRS)	-	-	-	-	-	-	-	-	-
Enhanced Data Rates for GSM Evolution (EDGE)	-	-	-	-	-	-	-	-	-
Code Division Multiple Access (CDMA)	-	-	-	-	-	-	-	-	-
Universal Mobile Telecommunications System (UMTS)	-	-	-	-	-	-	-	-	-
Wireless Local Area Network (WLAN)	-	-	-	-	-	-	-	K und A	x
								WPSInfo und Wi-fiP2P	x
Bluetooth		E	-	-	N	-	-	N	-
					Headset/A2DP			Health-Devices	
High Speed Packet Access (HSPA)		E	-	-	-	-	-	-	-
Near Field Communication (NFC)				E	N	-	-	N	K
					Tags				

Fortsetzung auf nächster Seite

Konnektivität	Plattform-Version								
	1.6	2.0	2.1	2.3	3.0	3.1	3.2	4.0	4.1
Portable Hotspot				E	-	-	-	-	-
Long Term Evolution (LTE)					E	-	-	-	-
Push-to-talk						E	-	-	-
Realtime transport protocol (RTP)						E	-	-	-
Voice over IP (VOIP)									
Android Beam								E	N
									Große Payloads
Netzwerkverbrauch								E	K
									Check metered network
Wi-Fi Direct								E	A
Network service discovery									E

Tabelle 3.8.: Konnektivität Bewertung

Multimedia	Plattform-Version								
	1.6	2.0	2.1	2.3	3.0	3.1	3.2	4.0	4.1
Aufnahme von Audio-Rohdaten	-	-	-	K	-	-	-	-	-
				Channels + BitRate (2.2)					
Media codecs	-	-	-	K	K	K	-	K	N
									low level media codecs
Musical Instrument Digital Interface (MIDI)	-	-	-	-	-	-	-	-	-
Wiedergabe von Audio-Rohdaten	-	-	-	-	-	-	-	-	-
Audioeffekte				E	-	-	-	-	-
HTTP Live Stream					E	-	-	K	K
Audioaufnahme ab Signal									E
Lückenlose Wiedergabe									E
Timed Text Tracks									E

Tabelle 3.9.: Multimedia Bewertung

Sensoren	Plattform-Version								
	1.6	2.0	2.1	2.3	3.0	3.1	3.2	4.0	4.1
Annäherungssensor	-	-	-	-	-	-	-	-	-
Drucksensor	-	-	-	-	-	-	-	-	-
Global Positioning System (GPS)	-	-	-	-	-	-	-	-	-
Gyroskop	-	-	-	-	-	-	-	-	-
Helligkeit	-	-	-	-	-	-	-	-	-
Höhensensor	-	-	-	-	-	-	-	-	-
Kompass	-	-	-	-	-	-	-	-	-
Accelerometer				E	-	-	-	-	-
Gravitation				E	-	-	-	-	-
Feuchtigkeit								E	-
Temperatur								E	-

Tabelle 3.10.: Sensoren Bewertung

Textbearbeitung	Plattform-Version								
	1.6	2.0	2.1	2.3	3.0	3.1	3.2	4.0	4.1
Autovervollständigung	-	-	-	-	-	-	-	-	-
Copy and Paste	-	-	-	-	N	-	-	-	K
					komplexe Daten				Styl. Text
Markieren	-	-	-	-	-	-	-	-	-
Speech to Text				K	-	-	-	K	-
				E(2.2)					
Drag and Drop					E	-	-	-	-
Rechtschreibkorrektur					E	-	-	-	-

Tabelle 3.11.: Textbearbeitung Bewertung

User Interface	Plattform-Version								
	1.6	2.0	2.1	2.3	3.0	3.1	3.2	4.0	4.1
Location Manager	-	-	-	A	-	-	-	-	-
				Location Updates					
Notifications	-	-	-	-	A	-	-	-	A
					Notification Builder				Big view
Text to Speech	E	-	-	K	K	-	-	A	-
								Multiple Text to Speech Engine	
Verschiedene Bildschirmauflösungen	E	-	-	-	A	-	A	K	-
					Tablets		720p		
Online-Konten Verwaltung		E	-	-	-	-	-	-	-
Quick Contact		E	-	-	-	-	-	-	-
Backup Service				E	-	-	-	-	-
Car Mode				E	-	-	-	-	-
Cloud to Device Messaging (C2DM)				E	-	-	-	-	-
Multitouch				E	-	-	-	-	-
Night Mode				E	-	-	-	-	-

Fortsetzung auf nächster Seite

User Interface	Plattform-Version								
	1.6	2.0	2.1	2.3	3.0	3.1	3.2	4.0	4.1
Grafikbeschleunigung					E	-	-	-	-
Action Bar					E	-	-	-	-
Unterstützt 720p (Fernseher)							E	-	-

Tabelle 3.12.: User Interface Bewertung

3.1. Allgemeiner Eindruck

Bis Plattform-Version 2.3 werden, im Vergleich zu den folgenden Versionen, viele Komponenten und die dazugehörigen Technologien nicht unterstützt. Insbesondere sind davon Komponenten aus den Tabellen User-Interface, Textbearbeitung und Konnektivität betroffen. Eine Ausnahme stellen die Plattform-Versionen 3.x dar, da diese ausschließlich für Tablets konzipiert sind. Erst ab Version 4.0 werden wieder sowohl Tablets als auch Smartphones unterstützt. Des Weiteren gibt es mit dieser Version unter anderem einige Verbesserungen in Verbindung mit der Kamera. Die höchste der untersuchten Versionen ist die Version 4.1. In dieser Version gibt es sehr viele kleine Änderungen aber auch Einführungen. Insbesondere gab es in den Bereichen Multimedia und App Komponenten viele Einführungen.

App Komponenten - Seit API-Level 4 können Apps Inhalte für das globale Suchmenü anbieten, dadurch kann der User gesuchte Inhalte finden ohne die App direkt aufrufen zu müssen. Eine Backup Funktion und Live-Wallpaper wurden mit API-Level 8 eingeführt. Durch die Backup Funktion können Daten einer App in der Cloud gespeichert werden und nach der Neuinstallation der App wieder hergestellt werden. API-Level 16 und höher bietet die Möglichkeit einen Service mit einer eigenen User-ID auszuführen.

Benutzerfreundlichkeit - Seit API-Level 4 ist es möglich Strings in verschiedenen Sprachen vorlesen zu lassen. Weiter wurden mit dem API-Level 4 Wischgesten eingeführt. API-Level 11 und höher bieten die Möglichkeit Drag and Drop zu nutzen.

Eingabegeräte - Seit API-Level 12 ist es möglich verschiedenen Eingabegeräte wie eine Tastatur, eine Maus oder einen Spiele-Controller zu nutzen. Vibrierende Eingabegeräte werden seit API-Level 16 unterstützt.

Enterprise - Im Enterprise-Bereich wurden innerhalb der betrachteten Versionen einige Neuentwicklungen getätigt, u.a. wurden Mobile-Device-Management- und Zertifikatmanagement-Funktionen hinzugefügt, außerdem werden ab Version 1.6 Virtual Private Networks (VPN) unterstützt.

Kamera - Die Kamera ist in allen betrachteten Android-Versionen unterstützt, und hat insbesondere ab Version 4.0 einige für den Benutzer deutlich sichtbare Updates erhalten, u.a. können nun während der Videoaufnahme Fotos gemacht werden.

Konnektivität - Die meisten grundlegenden Mobilfunkstandards werden in allen betrachteten Versionen unterstützt (z.B. GSM, CDMA). Ab Version 2.0 wird der schnellere Übertragungsstandard HSPA unterstützt, ab Version 3.0 dann der noch schnellere LTE Standard.

Multimedia - Die Funktionen, um Audiodaten zu verarbeiten sind schon seit API-Level 4 vorhanden. In API-Level 16 sind dann noch Extras hinzugekommen, mit welchen mehr Informationen zur Verfügung gestellt werden können (wie Timed Text Tracks) und eine individuellere Verarbeitung stattfinden kann.

Sensoren - Die meisten Sensoren wurden schon vor API-Level 4 unterstützt und wurden auch um keine Funktionen mehr erweitert. Der Accelerometer und der Gravitationsensor

kamen in API-Level 9 hinzu. Temperatur- und Feuchtigkeitssensoren wurden in API-Level 15 mit aufgenommen.

Textbearbeitung - Seit API-Level 8 ist es möglich Speech to Text zu nutzen. Drag and Drop kam erst mit API-Level 11 hinzu, hier wurde auch die Rechtschreibkorrektur eingeführt.

User Interface - Das User Interface wurde kontinuierlich erweitert. Ab API-Level 5 werden verschiedene Bildschirmgrößen unterstützt. Mit der Erweiterung für Tablets wurden diese Features nochmalig angepasst und danach im API-Level 13 um 720p erweitert. Um einen schnelleren Zugriff auf die jeweiligen Kontakte und die verschiedenen Möglichkeiten zur Kontaktaufnahme zu ermöglichen wurde im API-Level 6 Quick Contact eingeführt. In diesem Level wurde auch die Online-Konten-Verwaltung aufgenommen. Mit dem API-Level 9 wird neben kleinen Änderungen auch das Cloud to Device Management und der Backup Service eingeführt. Die Grafikbeschleunigung und die Action Bar gibt es ab dem API-Level 11.

4. Fazit und Ausblick

Ein Entwickler, der sich dafür entscheidet eine Android-App zu programmieren, sollte sich unter anderem mit folgenden Überlegungen auseinandersetzen. Er sollte sich über die Features, die einen wesentlichen Bestandteil der App ausmachen, klar werden. Auf diese kann eventuell nicht verzichtet werden, daher können sie zu sogenannten K.O.-Kriterien werden. Sobald er sich seiner Anforderungen bezüglich der Plattform-Version bewusst ist, kann er mit obigen Tabellen in 2.9, eine Mindestversion auswählen. Danach kann er sich überlegen, welche Plattform-Version die Zielversion darstellen soll. Hierbei sollte er sich mit der aktuellen Verteilung der Plattform-Versionen auf dem Markt beschäftigen, um möglichst viele potenzielle Nutzer zu erreichen. Des Weiteren sollte er sich einen Überblick über die für ihn am besten geeignete Version verschaffen, dies kann mithilfe der Tabellen in 3 geschehen.

Im Allgemeinen ist hervorzuheben, dass mit Version 2.3 und höher einige Verbesserungen kamen. Auch die Verteilung der Plattform-Versionen auf dem Markt zeigt zum jetzigen Zeitpunkt (Stand: 03.01.2013), dass sich die Entwicklung einer App erst ab dieser Version empfiehlt, da nur noch 10,4 Prozent der Nutzer die Plattform-Versionen 2.2 und niedriger verwenden, aber immerhin 50,8 Prozent noch die Version 2.3 besitzen.

Schon die Version 1.6 unterstützt die Barrierefreiheit, verschiedene Bildschirme und VPN. Im Vergleich zu den vorherigen Versionen, welche nicht Bestandteil dieser Studie waren, wurde 1.6 auch für die Entwickler optimiert. Die darauf folgende Version 2.0 zeichnet sich, aus Sicht des Benutzers, vor allem durch die verbesserte Bedienbarkeit aus. Dies wurde zum größten Teil mit Aktualisierungen realisiert, wobei sich wenig neue Funktionen ergaben. In der Plattform-Version 2.1 wurde der Fokus auf die Speicheroptimierung gelegt. Mit der Version 2.3 wurde die Benutzerschnittstelle verbessert. Der Versionssprung zu 3.0 brachte die Optimierung für Tablets mit sich. Darüber hinaus gab es eine Verbesserung der Konnektivität und die neue Komponente der 2D und 3D Animation. Des Weiteren gab es eine Erweiterung bezüglich der Enterprise Features. Mit der darauf folgenden Version 3.1 wird nun auch USB unterstützt. Mit dieser Version gibt es auch eine Optimierung der Benutzerschnittstelle. Der Schwerpunkt der Version 3.2 liegt auf der Unterstützung verschiedener Bildschirmgrößen und deren Auflösung. Die Version 4.0 bringt eine verbesserte Benutzeroberfläche und Konnektivität mit sich. Darüber hinaus wurden die Funktionen der Versionen 3.x, welche für Tablets vorgesehen waren, für Smartphones angepasst. Die in diesem Bericht betrachtete neuste Version 4.1 unterstützt externe Wiedergabegeräte, neue Technologien für eine flüssige Bedienung sowie einen Dienst um kurze Nachrichten zu verschicken. Des Weiteren wurde das Look and Feel der API verbessert.

Die Plattform-Version 4.2, welche im November 2012 erschienen ist, bringt eine Optimierung

4. Fazit und Ausblick

der Performance, eine verbesserte Benutzerschnittstelle sowie einige neue Features für Benutzer und Entwickler mit sich. Dazu gehören unter anderem die Quick Settings (ein schnellerer Zugriff auf die Einstellungen), mehrere Benutzerkonten auf Tablets, Unterstützung externer Bildschirme sowie RTL (Right to left) und eine "built in" Option für Entwickler (Möglichkeit auf jedem Android-Gerät zu debuggen) [And12].

Im Internet wird schon jetzt über die nächste Plattform-Version spekuliert. Das einzige was sicher zu sein scheint ist der Name der Version: "Key Lime Pie". Android verfolgt also weiterhin die bekannte Namensgebung, welche darauf beruht die Versionen nach Süßigkeiten zu benennen und dies in alphabetischer Reihenfolge [Tam12]. Ein Zweig aus dem git-Repository der Plattform Entwickler deutet daraufhin, dass es ein Update des Linux-Kernel auf die Version 3.8 geben wird [Fun13]. Des Weiteren befinden sich viele Spekulationen im Netz über lang ersehnte Features, wie zum Beispiel die Unterstützung von Orts- und NFC basierte Profile und die Zusammenlegung aller Messenger-Dienste von Google [Dou13].

A. Anhang

Listing A.1 MainActivity.java

```
/*
 * Copyright 2013 Julia Holzschuh, Ehssan Doust, Paul Metzger
 *
 * This file is part of Farbmischer
 *
 * Farbmischer is free software: you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * KiWe is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with Farbmischer. If not, see <http://www.gnu.org/licenses/>.
 */
package de.farbenmischer;

import android.annotation.SuppressLint;
import android.app.Activity;
import android.graphics.Color;
import android.os.Bundle;
import android.text.AndroidCharacter;
import android.view.Menu;
import android.view.View;
import de.farbenmischer.listener.ColorClickListener;
import de.farbenmischer.listener.ColorDragListener;
import de.farbenmischer.listener.ColorTouchListener;
import de.farbenmischer.utils.ColorHelper;

/**
 * Die einzige Activity die von Farbmischer genutzt wird. Stellt die Funktionalitaet bereit.
 */
public class MainActivity extends Activity {

    private static View farbeimer;
    private static boolean dragAndDropEnabled = false;

    public static View getFarbeimer() {
        return farbeimer;
    }

    public static boolean getDragAndDropEnabled() {
        return dragAndDropEnabled;
    }

    public static void setFarbeimer(View farbeimer) {
        MainActivity.farbeimer = farbeimer;
    }
}
```

Listing A.2 MainActivity.java

```
}

public static void setDragAndDropEnabled(boolean value) {
    MainActivity.dragAndDropEnabled = value;
}

/**
 * Wird beim Start der Anwendung ausgefuehrt.
 */
@SuppressWarnings("NewApi")
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    //Laed die UI
    setContentView(R.layout.activity_main);

    //Pruefe ob Drag and Drop von der Plattformversion unterstuetzt wird.
    int currentAPIVersion = android.os.Build.VERSION.SDK_INT;
    if (currentAPIVersion >= android.os.Build.VERSION_CODES.HONEYCOMB) {
        setDragAndDropEnabled(true);
    }

    findViewById(R.id.resetButton).setOnClickListener(new
        android.view.View.OnClickListener() {

        @Override
        public void onClick(View v) {
            ColorHelper.setFarbeimerColor(Color.WHITE);
        }
    });

    setFarbeimer(findViewById(R.id.mischung));

    //Setzt die entsprechenden Listener, abhaengig davon Drag and Drop
    unterstuetzt wird.
    if (getDragAndDropEnabled()) {
        getFarbeimer().setOnDragListener(new ColorDragListener());
        findViewById(R.id.rot).setOnTouchListener(
            new ColorTouchListener(Color.rgb(255, 0, 0)));
        findViewById(R.id.gelb).setOnTouchListener(
            new ColorTouchListener(Color.rgb(0, 255, 0)));
        findViewById(R.id.blau).setOnTouchListener(
            new ColorTouchListener(Color.rgb(0, 0, 255)));
    } //Wird ausgefuehrt wenn Drag and Drop nicht unterstuetzt wird.
    else {
        findViewById(R.id.rot).setOnClickListener(
            new ColorClickListener(Color.rgb(255, 0, 0)));
        findViewById(R.id.gelb).setOnClickListener(
            new ColorClickListener(Color.rgb(0, 255, 0)));
        findViewById(R.id.blau).setOnClickListener(
            new ColorClickListener(Color.rgb(0, 0, 255)));
    }
}
```

Listing A.3 MainActivity.java

```
        findViewById(R.id.mischung).setOnClickListener(new
            android.view.View.OnClickListener() {

                @Override
                public void onClick(View v) {
                    ColorHelper.addFarbeimerColor(ColorHelper.clickedColor);
                    ColorHelper.clickedColor = Color.TRANSPARENT;
                }
            });
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.activity_main, menu);
        return true;
    }
}
```

Listing A.4 ColorHelper.java

```
/*
 * Copyright 2013 Julia Holzschuh, Ehssan Doust, Paul Metzger
 *
 * This file is part of Farbmischer
 *
 * Farbmischer is free software: you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * KiWe is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with Farbmischer. If not, see <http://www.gnu.org/licenses/>.
 */
package de.farbenmischer.utils;

import android.graphics.Color;
import de.farbenmischer.MainActivity;

/**
 * Hilfsklasse fuer den Umgang mit Farben.
 */
public class ColorHelper {

    public static int clickedColor;

    private static int farbeimerColor;

    public static int getFarbeimerColor() {
        return farbeimerColor;
    }

    /**
     * Setzt die Farbe des Feldes in dem gemischt wird.
     */
    public static void setFarbeimerColor(int color) {
        farbeimerColor = color;
        MainActivity.getFarbeimer().setBackgroundColor(color);
    }

    /**
     * Mischt Farb zu dem Feld in dem Farben gemischt werden.
     */
    public static void addFarbeimerColor(int color) {
        setFarbeimerColor(ColorHelper.add(getFarbeimerColor(), color));
    }
}
```

Listing A.5 ColorHelper.java

```
}

/**
 * Addiert Farben und gibt die resultierende Farbe zurueck
 */
public static int add(int a, int b) {
    int aRed = Color.red(a);
    int aGreen = Color.green(a);
    int aBlue = Color.blue(a);

    int bRed = Color.red(b) / 10;
    int bGreen = Color.green(b) / 10;
    int bBlue = Color.blue(b) / 10;

    int newRed = (aRed + bRed) % 256;
    int newGreen = (aGreen + bGreen) % 256;
    int newBlue = (aBlue + bBlue) % 256;

    return Color.rgb(newRed, newGreen, newBlue);
}
}
```

Listing A.6 ColorClickListener.java

```
package de.farbenmischer.listener;

import de.farbenmischer.utils.ColorHelper;
import android.content.DialogInterface;
import android.content.DialogInterface.OnClickListener;
import android.view.View;

public class ColorClickListener implements android.view.View.OnClickListener {

    private int colorCode;

    public ColorClickListener(int color) {
        this.colorCode = color;
    }

    @Override
    public void onClick(View arg0) {
        ColorHelper.clickedColor = colorCode;
    }

}
}
```

Listing A.7 ColorDragListener.java

```
/*
 * Copyright 2013 Julia Holzschuh, Ehssan Doust, Paul Metzger
 *
 * This file is part of Farbmischer
 *
 * Farbmischer is free software: you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * KiWe is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with Farbmischer. If not, see <http://www.gnu.org/licenses/>.
 */
package de.farbenmischer.listener;

import android.annotation.SuppressLint;
import android.graphics.Color;
import android.view.DragEvent;
import android.view.View;
import android.view.View.OnDragListener;
import de.farbenmischer.utils.ColorHelper;

/**
 * Drag Listener wird genutzt wenn Drag and Drop zur Verfuegung steht.
 */
public class ColorDragListener implements OnDragListener {

    /**
     * Wird ausgefuehrt wenn ein Feld angetippt, nicht losgelassen und gezogen wird.
     */
    @SuppressWarnings("NewApi")
    @Override
    public boolean onDrag(View v, DragEvent event) {

        String color = "0;77;123";
        if (event != null && event.getClipData() != null
            && event.getClipData().getItemCount() > 0) {
            color = (String) event.getClipData().getItemAt(0).getText();
        }
        View view = (View) event.getLocalState();
        //Verschiedene Moeglichkeiten zu verschiedenen Zeitpunkten auf Drag and Drop
        zu reagieren
        switch (event.getAction()) {
            case DragEvent.ACTION_DRAG_STARTED:
                break;
            case DragEvent.ACTION_DRAG_ENTERED:

```

Listing A.8 ColorDragListener.java

```
        break;
//Nutzer hat das drop-target verlassen.
case DragEvent.ACTION_DRAG_EXITED:
    if (view != null)
        view.setVisibility(View.VISIBLE);
    break;
//Farbe wird ueber dem Farbeimer los gelassen.
case DragEvent.ACTION_DROP:
    if (view != null)
        view.setVisibility(View.VISIBLE);

    String[] splittedColor = color.split(";");

    int c = Color.argb(255, Integer.parseInt(splittedColor[0]),
        Integer.parseInt(splittedColor[1]),
        Integer.parseInt(splittedColor[2]));

    ColorHelper.addFarbeimerColor(c);

    break;
//Nutzer hat aufgehoeert Drag and Drop zu nutzen.
case DragEvent.ACTION_DRAG_ENDED:
    if (view != null)
        view.setVisibility(View.VISIBLE);
    break;
default:
    if (view != null)
        view.setVisibility(View.VISIBLE);
    break;
}
return true;
}
}
```

Listing A.9 ColorTouchListener.java

```
/*
 * Copyright 2013 Julia Holzschuh, Ehssan Doust, Paul Metzger
 *
 * This file is part of Farbmischer
 *
 * Farbmischer is free software: you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * KiWe is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with Farbmischer. If not, see <http://www.gnu.org/licenses/>.
 */
package de.farbenmischer.listener;

import android.content.ClipData;
import android.graphics.Color;
import android.view.MotionEvent;
import android.view.View;
import android.view.View.DragShadowBuilder;
import android.view.View.OnTouchListener;

/**
 * Touchlistener wird genutzt wenn Drag and Drop nicht zur Verfuegung steht
 */
public class ColorTouchListener implements OnTouchListener {

    private int colorCode;

    public ColorTouchListener(int color) {
        this.colorCode = color;
    }

    /**
     * Wird bei antippen ausgefuehrt.
     */
    @Override
    public boolean onTouch(View view, MotionEvent motionEvent) {
        if (motionEvent.getAction() == MotionEvent.ACTION_DOWN) {
            String color = Color.red(colorCode) + ";" + Color.green(colorCode)
                + ";" + Color.blue(colorCode);
            ClipData data = ClipData.newPlainText("COLOR", color);
            DragShadowBuilder shadowBuilder = new View.DragShadowBuilder(view);
            view.startDrag(data, shadowBuilder, view, 0);
            // view.setVisibility(View.INVISIBLE);
        }
    }
}
```

Listing A.10 ColorTouchListener.java

```
        return true;
    } else {
        return false;
    }
}
}
```

Listing A.11 activity_main.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/description"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/description"
        android:layout_marginTop="10dp" />

    <SurfaceView
        android:id="@+id/rot"
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:background="#ffff0000"
        android:layout_below="@+id/description"
        android:layout_marginLeft="20dp"
        android:layout_marginRight="20dp"
        android:layout_marginTop="20dp" />

    <SurfaceView
        android:id="@+id/blau"
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:background="#ff0000ff"
        android:layout_alignBottom="@+id/rot"
        android:layout_toRightOf="@+id/rot"
        android:layout_marginRight="20dp"
        android:layout_marginTop="20dp" />

    <SurfaceView
        android:id="@+id/gelb"
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:background="#ffffff00"
        android:layout_alignBottom="@+id/rot"
        android:layout_toRightOf="@+id/blau"
        android:layout_marginRight="20dp"
        android:layout_marginTop="20dp" />

    <SurfaceView
        android:id="@+id/mischung"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:background="@drawable/drop_shape"
        android:layout_centerHorizontal="true"
```

Listing A.12 activity_main.xml

```
        android:layout_centerVertical="true" />

    <TextView
        android:id="@+id/misch_label"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/misch_label"
        android:layout_below="@+id/mischung"
        android:layout_alignLeft="@+id/mischung" />

    <Button
        android:id="@+id/resetButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentEnd="true"
        android:layout_marginBottom="20dp"
        android:layout_marginRight="20dp"
        android:text="@string/reset" />
```

Literaturverzeichnis

- [And12] Android. Entwickler-Homepage. "<http://developer.android.com/index.html>", 2012. "[Online; Zugriff 20.12.2012]". (Zitiert auf den Seiten 12, 21 und 50)
- [AS12] A. H. und Andrei Socaciu. Fragmentierung auf der Android-Plattform. "<http://it-republik.de/jaxenter/artikel/Fragmentierung-auf-der-Android-Plattform-4737.html>", 2012. "[Online; Zugriff 10.01.2013]". (Zitiert auf Seite 28)
- [Dou13] P. Douglas. Android 5.0 Key Lime Pie release date, news and rumours. "<http://www.techradar.com/news/phone-and-communications/mobile-phones/android-5-0-key-lime-pie-release-date-news-and-rumours-1091500>", 2013. "[Online; Zugriff 19.04.2013]". (Zitiert auf Seite 50)
- [FH08] P. Fischer, P. Hofer. *Lexikon der Informatik*. Springer-Verlag, 2008. (Zitiert auf Seite 11)
- [Fun13] L. Funk. Android 5.0: Hinweise auf „Key Lime Pie“ mit Linux-Kernel 3.8. "<http://www.androidnext.de/news/android-5-0-linux-kernel-3-8-key-lime-pie/>", 2013. "[Online; Zugriff 02.04.2013]". (Zitiert auf Seite 50)
- [Tam12] A. Tamannai. Key Lime Pie: Name der nächsten Android-Iteration im Comic bestätigt. "<http://www.androidnext.de/news/key-lime-pie-name-der-naechsten-android-iteration-im-comic-bestaetigt/>", 2012. "[Online; Zugriff 02.04.2013]". (Zitiert auf Seite 50)

Alle URLs wurden zuletzt am 02.04.2013 geprüft.

Erklärung

Wir versichern, diese Arbeit selbstständig verfasst zu haben. Wir haben keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Wir haben diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Ort, Datum, Unterschrift

Ort, Datum, Unterschrift

Ort, Datum, Unterschrift