

Analyse und Evaluierung von plattformübergreifenden Spiel-Engines und Frameworks, anhand der Implementierung einer mobilen Beispielapplikation

Bachelor-Thesis

zur Erlangung des akademischen Grades B.Sc.

Sebastian Bohn

2036605



Hochschule für Angewandte Wissenschaften Hamburg
Fakultät Design, Medien und Information
Department Medientechnik

Erstprüfer: Prof. Dr. Edmund Weitz

Zweitprüfer: Prof. Dr. Andreas Pläß

vorläufige Fassung vom 25. November 2015

Inhaltsverzeichnis

1	Einleitung	6
1.1	Motivation	6
1.2	Gliederung	6
2	Mobile Systeme	7
2.1	Aktuelle Systeme auf dem mobilen Markt	7
2.1.1	iOS	7
2.1.2	Android	8
2.1.3	Windows Phone	8
2.2	Bedarfsanalyse	9
2.2.1	Markt- und Useranteile der jeweiligen Systeme	9
2.2.2	Verfügbare Applikationen / Games der Stores	9
3	Native Softwareentwicklung	10
3.1	Systemvoraussetzungen	10
3.1.1	iOS	10
3.1.2	Android	10
3.1.3	Windows Phone	10
3.2	SDKs und Versionen	10
3.2.1	iOS Versionen	11
3.2.2	Android Versionen	13
3.2.3	Windows Phone Versionen	14
3.3	Programmiersprachen	14
3.3.1	iOS	14
3.3.2	Android	14
3.3.3	Windows Phone	14
3.4	Entwicklungsumgebungen	14
3.4.1	iOS	14
3.4.2	Android	14
3.4.3	Windows Phone	14
4	Cross-Platform Entwicklung	15
4.1	Ziel	15
4.2	Funktionsweise	15
4.2.1	Technik	15
4.2.2	Geteilter Content	15

4.2.3	Übersetzung in die jeweiligen System	15
5	Cross-Platform Frameworks	16
5.1	Tools und Anbieter zur Entwicklung	16
5.2	Verweis auf Bachelorarbeit: „Plattformabhängige und –unabhängige Entwicklung mobiler Anwendungen am Beispiel von Geo-Wikipedia-App“	16
5.3	Gamespezifische Frameworks und Engines	16
5.3.1	Monogame	16
5.3.2	Cocos2D	16
5.3.3	Libgdx	16
5.3.4	Unity	16
5.3.5	Unreal Engine	16
5.3.6	Weitere Frameworks	16
5.4	Entwicklungsumgebungen	16
5.4.1	Unterstützte IDEs	16
5.4.2	Systembedingte Einschränkungen	16
6	Gegenüberstellung der Frameworks	17
6.1	Zielpattformen	17
6.2	Programmiersprachen	17
6.3	Unterstützung von 2D und 3D	17
6.4	Zugriff auf Hardware	17
6.5	Free- und Pro- Versionen	17
6.6	Einfluss auf Einstellungen	17
6.7	Zusätzlich benötigte Software	17
6.8	Aktualität - Versionen - Community	17
6.9	Zukunftsansichten	17
7	Analyse der Marktanteile	18
7.1	Menge an Firmen und Entwicklern	18
7.2	Menge an Games	18
8	Kosten-Nutzen Vergleich	19
9	Konzeption einer Applikation	20
9.1	Ideen	20
9.2	Anforderungen	20
9.3	User Stories	20
10	Implementierung der Applikation	21
10.1	Verwendete Frameworks und Engines	21
10.2	Verwendete APIs und SDKs	21
10.3	Assets und deren Verwendung	21

Inhaltsverzeichnis

11 Analyse messbarer Metriken	22
12 Vergleich der Messprotokolle	23
13 Fazit	24
Abbildungsverzeichnis	25
Tabellenverzeichnis	26
Literaturverzeichnis	27

Abstract

Zusammenfassung

1 Einleitung

1.1 Motivation

1.2 Gliederung

2 Mobile Systeme

2.1 Aktuelle Systeme auf dem mobilen Markt

2.1.1 iOS

iOS ist das mobile Betriebssystem des Unternehmen Apple. Dieses ist ein Derivat von Mac OS X, welches selbst auf Unix basiert. Es wird ausschließlich von Apple entwickelt und ist somit nur auf den eigenen Geräten iPhone, iPad und iPod touch zu finden. Mit der Entwicklung wurde unter externer und interner Geheimhaltung 2005 begonnen und das Resultat der Öffentlichkeit zum ersten Mal Anfang 2007 vorgestellt. Bis zur Version 4.0 wurde iOS mit dem Namen iPhoneOS betitelt. Das Konzept und Design ist schwerpunktmäßig auf hohe Benutzerfreundlichkeit und Funktionalität ausgelegt.

Durch die geschlossene Struktur des Systems sind eigene Derivate nicht möglich. Benutzer sind für den offiziellen Bezug von Applikationen auf Apples App Store angewiesen. Bei der Wahl der Hardware ist man auf die Produktpalette von Apple angewiesen, welche jährlich eine neue Generation veröffentlicht. Die Personalisierung der Geräte ist nur bedingt möglich, da Anbieter von Drittsoftware keinen Zugriff auf das System haben und Anwendungen nur offiziell über den eigenen App Store bezogen werden können. Dies bietet jedoch den Vorteil einer Qualitätssicherung durch Apple, da Applikationen vor der Veröffentlichung einer Prüfung unterzogen werden.

Vorteile:

- Kompatibilität von Software und Hardware
- Benutzerfreundlichkeit
- Geräteübergreifende Kommunikation
- Kontrollen bei Veröffentlichung von Anwendungen

Nachteile:

- Restriktive Firmenpolitik
- Proprietäres System
- Hardwareauswahl
- Anwendungen nur über den App Store

2.1.2 Android

Android ist ein Open Source Betriebssystem und gleichzeitig eine Software-Plattform, welches stark im mobilen Bereich vertreten ist und auf dem Linux-Kernel basiert. Zu finden ist diese auf Smartphones, Tablet-Computern, Netbooks und auch auf Smart-TV Geräten. ([Open Handset Alliance - Android Overview 2015](#)) Entwickelt wird Android von der Open Handset Alliance (OHA), welche von Google gegründet wurde. Die OHA wurde im November 2007 gegründet und ist ein Konsortium von mehr als 80 Unternehmen aus den Bereichen Mobilfunknetz, Geräteherstellung, Halbleiterindustrie, Marketing und Software. ([Open Handset Alliance - Alliance Members 2015](#)) Der Grund für die Entwicklung von Android war und ist es, einen offenen Standard für mobile Geräte zu schaffen. ([Open Handset Alliance - Alliance Overview 2015](#)) Durch seine Offenheit ermöglicht Android Entwicklern große Freiheit bei der Programmierung von Applikationen. Eigene Entwicklungen können auch mit Anwendungen von Google, wie zum Beispiel Google Maps, verknüpft werden. Auch der Hardwarebereich bietet ein breites Spektrum an Geräten mit kostengünstigen, bis hochpreisigen Angeboten, sowohl mit einfacher bis qualitativ hochwertiger, technischer Ausstattung. Benutzer haben die Möglichkeit, ihre Geräte weitestgehend frei zu gestalten und einzustellen. Bei der Installation von Anwendungen sind sie auch nicht zwangsläufig an einen Anbieter gebunden.

Vorteile:

- Open Source
- Unabhängigkeit von Anbietern
- Personalisierung
- Hardwareangebot

Nachteile:

- Hohe Verbreitung von Schadsoftware
- Aktualität der Version ist abhängig vom Gerätehersteller

2.1.3 Windows Phone

Entwickler Microsoft stellt seit dem Jahr 2000 Betriebssysteme für mobile Geräte her. ([Fran Berkman 2012](#)) Seitdem hat sich die Namensgebung von Windows Mobile, über Windows Phone, bis zum aktuellsten Windows 10 Mobile vorgearbeitet. Um im allgemeinen Bezug nicht zwischen den Namen hin und her zu wechseln, wird in dieser Arbeit, wenn mobile Windows Systeme erwähnt werden, der Name Windows Phone benutzt. Die frühen Versionen von Windows Phone, also Windows Mobile und Windows Phone 7 stammen noch von dem Windows CE Kernel ab, wobei die

aktuellen Versionen, Windows Phone 8 und Windows 10 Mobile, Derivate des Windows NT Kernels sind. Mit dem neuesten Ableger, Windows 10 Mobile, verspricht Microsoft eine homogene Kommunikations- und Anwendungsstruktur zwischen allen Geräten die mit diesem System betrieben werden. Dazu zählen nicht nur Smartphones und Tablets, sondern auch Notebooks, Desktop PCs und die Spielkonsole Xbox One. ([Microsoft 2015](#))

Microsoft verfolgt mit Windows Phone eine ähnlich geschlossene und proprietäre Struktur wie der Konkurrent Apple. Eigene Derivate des Systems sind also nicht offiziell möglich. Auch die Benutzer müssen für neue Anwendungen auf das Angebot des Windows Stores zurückgreifen. Jedoch will Microsoft Entwicklern die Möglichkeit bieten, zukünftige Anwendungen universell verfügbar zu machen, das diese auf allen Windows Systemen nutzbar sind. Microsoft arbeitet außerdem an einer Technik, die bestehende Android und iOS Anwendungen auf die Windows Plattform überführen kann. ([Golem 2015](#))

Die aktuellen Windows Phone Versionen sind durch eine Allianz von Windows und Nokia, hauptsächlich auf mobilen Geräten von Nokia zu finden. ([Microsoft 2014](#)) Aber auch andere Hersteller bieten Geräte mit Windows Phone, jedoch bisher in einem überschaubaren Umfang.

Vorteile:

- Kompatibilität von Software und Hardware
- Universelle Anwendungen
- Benutzerfreundlichkeit

Nachteile:

- Proprietäres System
- Anwendungen nur aus dem Windows Store
- Geringeres Angebot an Anwendungen

([Reddit 2015](#))

2.2 Bedarfsanalyse

2.2.1 Markt- und Useranteile der jeweiligen Systeme

2.2.2 Verfügbare Applikationen / Games der Stores

3 Native Softwareentwicklung

Softwareentwicklung für ein bestimmtes System wird als nativ (lat.: angeboren, natürlich) bezeichnet. Hier sind Dateiformate, Programmiersprachen, Hardware, Entwicklungsumgebungen und Kompilierung genau an die Zielplattform angepasst. Nativer Code ist in der Lage alle individuellen Eigenschaften einer Zielplattform anzusprechen, ohne dabei eine eventuelle Portierbarkeit zu berücksichtigen. ([John Daintith 2004](#)) Welche Anforderungen iOS, Android und Windows Phone bezüglich nativer Entwicklung voraussetzen, soll in diesem Kapitel näher erläutert werden.

3.1 Systemvoraussetzungen

Um eine Applikation für eine bestimmte Zielplattform zu entwickeln, ist es unter Umständen nötig die Voraussetzung eines bestimmten Betriebssystems zu erfüllen.

3.1.1 iOS

3.1.2 Android

Die Entwicklung von Android Applikationen ist an kein bestimmtes System gebunden. Somit lassen sich diese unter Windows, OS X und Linux Systemen entwickeln. Windows Benutzer sollten mindestens Windows XP nutzen. Darüber hinaus können alle aktuelleren Versionen genutzt werden, wobei alle 32-Bit Editionen unterstützt werden und ab Windows 7 auch die mit 64-Bit. Mac Systeme werden ab OS X 10.5.8 von den offiziellen Entwicklungswerkzeugen unterstützt. Um auf einem Linux System zu entwickeln, kann man dies beispielsweise unter Ubuntu ab Version 8.04 tun. Bei 64-Bit Versionen ist es notwendig, dass diese fähig ist 32-Bit Anwendungen auszuführen. Da die Auswahl an Linux-Distributionen sehr umfangreich ist, wird an dieser Stelle auf diese nicht weiter eingegangen. ([Sue Smith 2013](#))

3.1.3 Windows Phone

3.2 SDKs und Versionen

Software Development Kits, kurz SDKs, liefern dem Entwickler die Werkzeuge, Anwendungen und bestenfalls eine aktuelle Dokumentation, um für eine bestimmte Zielplattform zu entwickeln. Auch sind sie notwendig, um geschriebenen Code zu

interpretieren und kompilieren. Um die aktuellste Version eines mobilen Systems zu unterstützen, muss das SDK auf ebenso aktuellen Stand sein.

3.2.1 iOS Versionen

Apple nutzt für seine Produkte Codenamen, die keinem bestimmten Muster folgen. Verbrauchern sind diese meist unbekannt, da diese überwiegend intern genutzt werden.

Codename	Version 9 - 6	Erscheinungsdatum
Monarch	9.2 Beta	3. November 2015
Monarch	9.1	21. Oktober 2015
Monarch	9.0.x	16. September 2015
Copper	8.4.x	30. Juni 2015
Stowe	8.3	8. April 2015
OkemoZurs	8.2	9. März 2015
OkemoTaos	8.1.x	9. Dezember 2015
Okemo	8.0.x	17. September 2014
Sochi	7.1.x	10. März 2014
Innsbruck	7.0.x	18. September 2013
Brighton	6.1.x	21. Februar 2013
Sundance	6.0.x	19. September 2012

Tabelle 3.1: iOS Versionen 9 - 6 und ihr Erscheinungsdatum
([the iphone wiki 2015](#))

Codename	Version 5 - 1	Erscheinungsdatum
Hoodoo	5.1.x	7. März 2012
Telluride	5.0.x	12. Oktober 2011
Durango	4.3.x	9. März 2011
Jasper	4.2.x	22. November 2010
Baker	4.1	8. September 2010
Apex	4.0.x	21. Juni 2010
Wildcat	3.2.x	3. April 2010
Northstar	3.1.x	9. September 2009
Kirkwood	3.0.x	17. Juni 2009
Timberline	2.2.x	21. November 2008
Sugarbowl	2.1.x	9. September 2008
Big Bear	2.x	11. Juli 2008
Little Bear	1.1.x	14. September 2007
Alpine	1.0.x	29. Juni 2007

Tabelle 3.2: iOS Versionen 5 - 1 und ihr Erscheinungsdatum
([the iphone wiki 2015](#))

3.2.2 Android Versionen

Android Versionen sind nach süßen Leckereien benannt und dem Anfangsbuchstaben nach alphabetisch aufsteigend.

Codename	Version	API Level 23 - 1	Erscheinungsdatum
Marshmallow	6	23	5. Oktober 2015
Lollipop	5.1.x	22	9. März 2015
Lollipop	5.0.x	21	3. November 2014 - 19. Dezember 2014
Wear	4.4W	20	Juni 2014
KitKat	4.4.x	19	31. Oktober 2013 - 19. Juni 2014
Jelly Bean	4.3.x	18	24. Juli 2013 - 4. Oktober 2013
Jelly Bean	4.2.x	17	13. November 2012 - 12. Februar 2013
Jelly Bean	4.1.x	16	27. Juni 2012 - 10. Oktober 2012
Ice Cream Sandwich	4.0.3 - 4.0.4	15, NDK 8	16. Dezember 2011 - 4. Februar 2012
Ice Cream Sandwich	4.0 - 4.0.2	14, NDK 7	19. Oktober 2011 - 15. Dezember 2011
Honeycomb	3.2.x	13	16. Juli 2011
Honeycomb	3.1	12, NDK 6	10. Mai 2011
Honeycomb	3	11	23. Februar 2011
Gingerbread	2.3.3 - 2.3.7	10	23. Februar 2011 - 20. September 2011
Gingerbread	2.3 - 2.3.2	9, NDK 5	6. Dezember 2010 - Januar 2011
Froyo	2.2 - 2.2.2	8, NDK 4	20. Mai 2010 - Januar 2011
Eclair	2,1	7, NDK 3	12. Dezember 2010
Eclair	2.0.1	6	3. Dezember 2009
Eclair	2	5	26. Oktober 2009
Donut	1.6	4, NDK 2	15. September 2009
Cupcake	1.5	3, NDK 1	30. April 2009
ohne Codename	1.1	2	10. Februar 2009
ohne Codename	1	1	23. September 2008

Tabelle 3.3: Android Versionen und ihr Erscheinungsdatum
([Android Source - Codenames, Tags, and Build Numbers 2015](#), [Wikipedia - Liste von Android-Versionen 2015](#))

3.2.3 Windows Phone Versionen

3.3 Programmiersprachen

In der nativen Entwicklung werden für jede Zielplattform bestimmte Programmiersprachen unterstützt.

3.3.1 iOS

3.3.2 Android

Android Applikationen werden in Java entwickelt. Demnach ist es notwendig vorab eine aktuelle Java Version (JDK) zu installieren. Diese wird von dem Unternehmen Oracle, mit der aktuellen Version 8 Update 66, vertrieben. ([Oracle - Java SE 2015](#))

3.3.3 Windows Phone

3.4 Entwicklungsumgebungen

Für die Entwicklung werden jeweilig verschiedene IDEs (Integrated Development Environment) seitens der Betreiber unterstützt und empfohlen. Eine Besonderheit bei IDEs für mobile Systeme ist die Unterstützung eines Simulators. Dieser simuliert ein spezifiziertes Gerät auf virtuelle Weise, um Entwicklungen direkt testen zu können.

3.4.1 iOS

3.4.2 Android

Android empfiehlt das eigene Android Studio, welches die offizielle IDE für Android Entwicklung darstellt und zusätzlich das aktuelle SDK mitliefert. Android Studio basiert auf der IDE IntelliJ IDEA und ist frei verfügbar. ([Android Develop Tools 2015](#)) Alternativer Vorgänger ist die quelloffene IDE Eclipse.

3.4.3 Windows Phone

4 Cross-Platform Entwicklung

4.1 Ziel

4.2 Funktionsweise

4.2.1 Technik

4.2.2 Geteilter Content

4.2.3 Übersetzung in die jeweiligen System

5 Cross-Platform Frameworks

5.1 Tools und Anbieter zur Entwicklung

5.2 Verweis auf Bachelorarbeit: „Plattformabhängige und –unabhängige Entwicklung mobiler Anwendungen am Beispiel von Geo-Wikipedia-App“

5.3 Gamespezifische Frameworks und Engines

5.3.1 Monogame

5.3.2 Cocos2D

5.3.3 Libgdx

5.3.4 Unity

5.3.5 Unreal Engine

5.3.6 Weitere Frameworks

5.4 Entwicklungsumgebungen

5.4.1 Unterstützte IDEs

5.4.2 Systembedingte Einschränkungen

6 Gegenüberstellung der Frameworks

6.1 Zielplattformen

6.2 Programmiersprachen

6.3 Unterstützung von 2D und 3D

6.4 Zugriff auf Hardware

6.5 Free- und Pro- Versionen

6.6 Einfluss auf Einstellungen

6.7 Zusätzlich benötigte Software

6.8 Aktualität - Versionen - Community

6.9 Zukunftsaussichten

7 Analyse der Marktanteile

7.1 Menge an Firmen und Entwicklern

7.2 Menge an Games

8 Kosten-Nutzen Vergleich

9 Konzeption einer Applikation

9.1 Ideen

9.2 Anforderungen

9.3 User Stories

10 Implementierung der Applikation

10.1 Verwendete Frameworks und Engines

10.2 Verwendete APIs und SDKs

10.3 Assets und deren Verwendung

11 Analyse messbarer Metriken

12 Vergleich der Messprotokolle

13 Fazit

Abbildungsverzeichnis

Tabellenverzeichnis

3.1	iOS Versionen 9 - 6 und ihr Erscheinungsdatum	11
3.2	iOS Versionen 5 - 1 und ihr Erscheinungsdatum	12
3.3	Android Versionen und ihr Erscheinungsdatum	13

Literaturverzeichnis

Overview of Android by the Open Handset Alliance, http://www.openhandsetalliance.com/android_overview.html, letzter Zugriff: 24.11.2015

Overview of the Open Handset Alliance, http://www.openhandsetalliance.com/oha_overview.html, letzter Zugriff: 24.11.2015

Members of the Open Handset Alliance, http://www.openhandsetalliance.com/oha_members.html, letzter Zugriff: 24.11.2015

Codenames, Tags, and Build Numbers in the history of Android, <https://source.android.com/source/build-numbers.html>, letzter Zugriff: 24.11.2015

Übersicht von allen Android Versionen mit Veröffentlichungsdatum, https://de.wikipedia.org/wiki/Liste_von_Android-Versionen, letzter Zugriff: 24.11.2015

A Dictionary of Computing - native software, <http://www.encyclopedia.com/doc/1011-nativesoftware.html>, letzter Zugriff: 24.11.2015

Android SDK Requirements, <http://code.tutsplus.com/tutorials/android-sdk-requirements--mobile-20086>, letzter Zugriff: 24.11.2015

Firmware, <https://www.theiphonewiki.com/wiki/Firmware>, letzter Zugriff: 24.11.2015

Java SE Development Kit 8 Downloads, <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>, letzter Zugriff: 24.11.2015

Android Studio Overview, <http://developer.android.com/tools/studio/index.html>, letzter Zugriff: 24.11.2015

Reddit - Pros and cons of Windows phone, and why should I buy one instead of an android or a iPhone?, https://www.reddit.com/r/windowsphone/comments/3h21lf/pros_and_cons_of_windows_phone_and_why_should_i/, letzter Zugriff: 25.11.2015

Microsoft - Microsoft und Nokia Geräte, <https://www.microsoft.com/de-de/nokia.aspx>, letzter Zugriff: 25.11.2015

Literaturverzeichnis

Microsoft demonstriert Android- und iOS-Apps unter Windows, <http://www.golem.de/news/windows-10-microsoft-demonstriert-android-und-ios-apps-unter-windows-1504-113812.html>, letzter Zugriff: 25.11.2015

Microsoft - Windows 10 Features, <https://www.microsoft.com/de-de/windows/features>, letzter Zugriff: 25.11.2015

Microsoft Mobile: From Pocket PC to Windows Phone 8, <http://mashable.com/2012/10/29/microsoft-mobile-history/#DYxZxZ7wTuqD>, letzter Zugriff: 25.11.2015

Ich versichere, die vorliegende Arbeit selbstständig ohne fremde Hilfe verfasst und keine anderen Quellen und Hilfsmittel als die angegebenen benutzt zu haben. Die aus anderen Werken wörtlich entnommenen Stellen oder dem Sinn nach entlehnten Passagen sind durch Quellenangaben eindeutig kenntlich gemacht.

Ort, Datum

Sebastian Bohn