

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 15. 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 | 9 |
| 2.1.4 | Weitere Systeme | 9 |
| 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 in den jeweiligen Systemen | 10 |
| 3.1 | Hardwarevoraussetzungen | 10 |
| 3.2 | Programmiersprachen | 10 |
| 3.3 | Entwicklungsumgebungen | 10 |
| 4 | Stores für mobile Spiele | 11 |
| 4.1 | Allgemeine Bedingungen für Entwickler | 11 |
| 4.2 | Maximale App-Größe | 11 |
| 4.3 | Anforderungen an den Quellcode | 11 |
| 4.4 | Kosten und Abgaben | 11 |
| 5 | Cross-Plattform Entwicklung | 12 |
| 5.1 | Sinn und Gedanke von Cross-Plattform Entwicklung | 12 |
| 5.2 | Funktionsweise von Cross-Plattform Entwicklung | 12 |
| 5.2.1 | Technik | 12 |
| 5.2.2 | Geteilter Content | 12 |
| 5.2.3 | Übersetzung in die jeweiligen System | 12 |
| 6 | Cross-Plattform Frameworks | 13 |
| 6.1 | Tools und Anbieter zur Entwicklung | 13 |
| 6.2 | Verweis auf Bachelorarbeit: „Plattformabhängige und –unabhängige Entwicklung mobiler Anwendungen am Beispiel von Geo-Wikipedia-App“ | 13 |

| | | |
|-----------|--|-----------|
| 6.3 | Gamespezifische Frameworks und Engines | 13 |
| 6.3.1 | Monogame | 13 |
| 6.3.2 | Cocos2D | 13 |
| 6.3.3 | Libgdx | 13 |
| 6.3.4 | Unity | 13 |
| 6.3.5 | Unreal Engine | 13 |
| 6.3.6 | Weitere Frameworks | 13 |
| 6.4 | Entwicklungsumgebungen | 13 |
| 6.4.1 | Unterstützte IDEs | 13 |
| 6.4.2 | Systembedingte Einschränkungen | 13 |
| 7 | Gegenüberstellung der Frameworks | 14 |
| 7.1 | Zielpattformen | 14 |
| 7.2 | Skalierbarkeit der Menge der Plattformen | 14 |
| 7.3 | Programmiersprachen | 14 |
| 7.4 | Unterstützung von 2D und 3D | 14 |
| 7.5 | Zugriff auf Hardware | 14 |
| 7.6 | Free- und Pro- Versionen | 14 |
| 7.7 | Einfluss auf Einstellungen | 14 |
| 7.8 | Zusätzlich benötigte Software | 14 |
| 7.9 | Aktualität - Versionen - Community | 14 |
| 7.10 | Zukunftsansichten | 14 |
| 8 | Analyse der Marktanteile | 15 |
| 8.1 | Menge an Firmen und Entwicklern | 15 |
| 8.2 | Menge an Games | 15 |
| 9 | Kosten-Nutzen Vergleich | 16 |
| 10 | Grundgerüst und Aufbau eines Cross-Plattform Projekts | 17 |
| 10.1 | Geteilter Content | 17 |
| 10.2 | Plattformabhängiger Content | 17 |
| 10.3 | Grundaufbau bei Engines | 17 |
| 10.4 | Grundaufbau bei Frameworks | 17 |
| 11 | Game-typische Design Patterns und Architekturen | 18 |
| 11.1 | Architekturen | 18 |
| 11.2 | Patterns | 18 |
| 11.3 | Verweis auf Bachelorarbeit: „Use of Design Patterns for mobile game Development“ | 18 |
| 12 | Nutzen von Architekturen | 19 |
| 12.1 | Pro | 19 |
| 12.1.1 | Skalierbarkeit | 19 |

Inhaltsverzeichnis

| | |
|--|-----------|
| 12.1.2 Lesbarkeit | 19 |
| 12.1.3 Wiederverwertbarkeit | 19 |
| 12.2 Contra | 19 |
| 12.2.1 KISS - Keep it simple stupid | 19 |
| 13 Konzeption einer Applikation | 20 |
| 13.1 Ideen | 20 |
| 13.2 Anforderungen | 20 |
| 13.3 User Stories | 20 |
| 14 Implementierung der Applikation | 21 |
| 14.1 Verwendete Frameworks und Engines | 21 |
| 14.2 Verwendete APIs und SDKs | 21 |
| 14.3 Assets und deren Verwendung | 21 |
| 15 Analyse messbarer Metriken | 22 |
| 16 Vergleich der Messprotokolle | 23 |
| 17 Fazit | 24 |
| Abbildungsverzeichnis | 25 |
| Literaturverzeichnis | 26 |

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. Entwickler haben die Möglichkeit, durch das iOS Software Development Kit (SDK) eigene Applikationen zu entwickeln und in Apples App Store zu veröffentlichen. 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.

iOS Versionen im Überblick:

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. 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. Der Grund für die Entwicklung von Android war und ist es, einen offenen Standard für mobile Geräte zu schaffen.

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

Android Versionen im Überblick:

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

2.1.4 Weitere Systeme

2.2 Bedarfsanalyse

2.2.1 Markt- und Useranteile der jeweiligen Systeme

2.2.2 Verfügbare Applikationen / Games der Stores

3 Native Softwareentwicklung in den jeweiligen Systemen

3.1 Hardwarevoraussetzungen

3.2 Programmiersprachen

3.3 Entwicklungsumgebungen

4 Stores für mobile Spiele

4.1 Allgemeine Bedingungen für Entwickler

4.2 Maximale App-Größe

4.3 Anforderungen an den Quellcode

4.4 Kosten und Abgaben

5 Cross-Plattform Entwicklung

5.1 Ziel von Cross-Plattform Entwicklung

5.2 Funktionsweise von Cross-Plattform Entwicklung

5.2.1 Technik

5.2.2 Geteilter Content

5.2.3 Übersetzung in die jeweiligen System

6 Cross-Plattform Frameworks

6.1 Tools und Anbieter zur Entwicklung

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

6.3 Gamespezifische Frameworks und Engines

6.3.1 Monogame

6.3.2 Cocos2D

6.3.3 Libgdx

6.3.4 Unity

6.3.5 Unreal Engine

6.3.6 Weitere Frameworks

6.4 Entwicklungsumgebungen

6.4.1 Unterstützte IDEs

6.4.2 Systembedingte Einschränkungen

7 Gegenüberstellung der Frameworks

7.1 Zielplattformen

7.2 Skalierbarkeit der Menge der Plattformen

7.3 Programmiersprachen

7.4 Unterstützung von 2D und 3D

7.5 Zugriff auf Hardware

7.6 Free- und Pro- Versionen

7.7 Einfluss auf Einstellungen

7.8 Zusätzlich benötigte Software

7.9 Aktualität - Versionen - Community

7.10 Zukunftsaussichten

8 Analyse der Marktanteile

8.1 Menge an Firmen und Entwicklern

8.2 Menge an Games

9 Kosten-Nutzen Vergleich

10 Grundgerüst und Aufbau eines Cross-Plattform Projekts

10.1 Geteilter Content

10.2 Plattformabhängiger Content

10.3 Grundaufbau bei Engines

10.4 Grundaufbau bei Frameworks

11 Game-typische Design Patterns und Architekturen

11.1 Architekturen

11.2 Patterns

11.3 Verweis auf Bachelorarbeit: „Use of Design Patterns for mobile game Development“

12 Nutzen von Architekturen

12.1 Pro

12.1.1 Skalierbarkeit

12.1.2 Lesbarkeit

12.1.3 Wiederverwertbarkeit

12.2 Contra

12.2.1 KISS - Keep it simple stupid

13 Konzeption einer Applikation

13.1 Ideen

13.2 Anforderungen

13.3 User Stories

14 Implementierung der Applikation

14.1 Verwendete Frameworks und Engines

14.2 Verwendete APIs und SDKs

14.3 Assets und deren Verwendung

15 Analyse messbarer Metriken

16 Vergleich der Messprotokolle

17 Fazit

Abbildungsverzeichnis

Literaturverzeichnis

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