ulm university universität
uulm

Universität Ulm | 89069 Ulm | Germany

Fakultät für Ingenieurwissenschaften, Informatik und Psychologie
Institut für Datenbanken und Informationssysteme

Mobile Application Lab

Ausarbeitung zur App an der Universität Ulm

Vorgelegt von:
Fabian Fischbach, Luis Beaucamp und Tim Stenzel

Gutachter:
Marc Schickler

Betreuer:
Marc Schickler

2017

Fassung 24. März 2017

Satz: PDF-LaTeX 2$_\varepsilon$

Inhaltsverzeichnis

1 Einleitung

In dieser Dokumentation wird die Entwicklung einer App im Rahmen des Anwendungsfaches Mobile Application Lab an der Universität Ulm und die dadurch erstellte App in Form einer Quartett App vorgestellt.

1.1 Motivation und Problemstellung

Die Problemstellung wurde uns im Rahmen dieses Projektes schon gegeben, da wir uns auf die Entwicklung einer Quartett Apps für Smartphones konzentrieren sollten. Das Prinzip des beliebten Kartenspiels für ein Smartphone entwickelt werden und so zu jeder Zeit und an jedem Ort auch ganz ohne physiche Karten spielbar sein.

Beim Anschauen des aktuellen Marktes für Quartett Apps fält schnell die Vielzahl an verschiedenen Apps auf. Diese erweisen nach genauerem Betrachten doch erhbliche Mängel auf. So sind manche von ihnen sehr veraltet und funktionieren nicht mehr richtig auf neueren Smartphone Modellen. Auch entsprechen diese inhaltlich nicht unseren Vorstellungen von einer guten Quartett App. Sie sind sehr beschränkt, was die verschiedenen Spielmodi angeht und beschräken sich meistens auf ein einziges Kartendeck oder Decks aus einem Themengebiet.

1.2 Zielsetzung

Da auf dem Markt eine Nachfrage besteht wollen wir diese ausnutzen und eine eigene Quartett Anwendung erstellen. Diese wollen wir auf Basis von Android und Java ent-

wickeln. Dabei geht es uns primär darum, den Umgang mit den neuen Techniken zu erlernen und eine Grunderfahrung im Programmieren von Android Anwendungen zu erlangen, sodass wir diese nach Abschließen des Projektes beherrschen.

Inhaltlich möchten wir eine Quartett App entwickeln, die sich als Singleplayer wie ein richtiges Quartett Spiel spielen lässt. Sie soll verschiedene Spielmodi haben, welche frei konfigurierbar sein sollen. Zudem soll die Auswahl an Decks breit gefächert sein, was durch einen Deckcreator und einer Onlinefunktion zum Up- und Downloaden realisiert werden soll. Der Anwendung soll zudem die Möglichkeit bieten, alle Karten anzugucken und laufende Spiele zu unterbrechen. Dabei soll die App benutzerfreundlich sein und schön aussehen, sowie auf den neusten aber auch auf älteren Android Smartphones lauffähig sein.

1.3 Struktur der Arbeit

Beschreibe in diesem Abschnitt die Struktur der Arbeit!

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

2

Grundlagen

2.1 Quartettspiel

Zum Quartettspielen sind natürlich einige Regeln notwendig, die im Folgenden erklärt werden.
Gespielt wird Eins gegen Eins, Spieler gegen Computer. Zuerst wählt der Spieler den Schwierigkeitsgrad (leicht, mittel oder schwer) des Computers, dann den Spielmodus und das Limit für das Spielende (Zeit-, Runden- oder Punktemodus und entsprechend die Spielzeit, Runden- oder Punkteanzahl). Danach wird ein Deck gewählt, gemischelt, Computer und Spieler bekommen jeweils die Hälfte der Karten verdeckt auf einem Stapel, bei dem immer nur die oberste Karte sichtbar ist und es wird zufällig bestimmt wer mit dem ersten Zug beginnen darf. Ein Zug sieht immer folgendermaßen aus: Der Spieler, der am Zug ist, wählt ein Attribut (Beispiel: in einem Autoquartett die max. Geschwindigkeit) und nennt den entsprechenden Wert. Der andere Spieler nennt nun ebenfalls seinen Wert (Beispiel von oben: max. Geschwindigkeit) und die Werte werden verglichen. Für jedes Attribut wurde vor einem Spiel festgelegt ob für dieses ein höherer oder niedrigerer Wert gewinnt. Der Vergleich wird also durchgeführt. Der Spieler mit dem besseren Wert gewinnt den Vergleich, bekommt beide Karten der Runde unter seinen Stapel und darf im nächsten Vergleich das Attribut wählen. Bei einem Unentschieden behält jeder seine Karten, legt sie unter seinen Stapel und der Spieler, der das Attribut gewählt hat, wählt auch das Nächste.

Dies sind die normalen Regeln und Spielmodi für ein Quartettspiel. In unserer App gibt es aber zusätzlich neben dem normalen Modus (bei dem jeweils, wie festgelegt, der höhere oder niedrigere Wert den Vergleich gewinnt) auch noch den sog. Insanemodus, bei dem

jeweils nicht der vorher festgelegte höhere oder niedrigere Wert gewinnt, sondern genau umgekehrt. Damit es im Spiel aber nicht zu Verwirrungen kommt ist immer neben jedem Attribut ein Pfeil, der angibt, ob höher oder niedriger besser ist. Egal ob der normale Modus oder Insanemodus gewählt wird, kann man sich noch zusätzlich entscheiden, ob man den Expertenmodus spielen möchte oder nicht. Der Expertenmodus ist aber nichts für Anfänger, denn es werden die Werte einer Karte durch '?' ersetzt, sodass man das Deck bzw. die Karten schon ein bisschen besser kennen muss um hier erfolgreich zu sein.

2.2 Mobile Plattform

Android ist ein mobiles Betriebssystem, also für Smartphones und Tablets, das von Google entwickelt wurde und auf Linux basiert. Die App-Entwicklung ist geprägt durch einzelne Aktivitäten (ein angezeigter Screen ist eine Aktivität), die miteinander kommunizieren und in ihrer 'Lebenszeit' ein vorgegebenes Zustandmodell 2.1 durchlaufen.

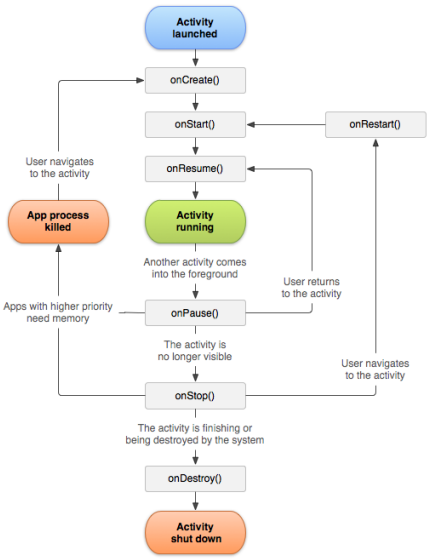

Abbildung 2.1: Android Zustandsmodell

Dieses Zustandmodell ist auch anfangs einer der Nachteile von Android, da es nicht so leicht zu verstehen ist und uns auch ein paar Probleme bereitet hat. Nachdem wir uns

aber im Laufe der App-Entwicklung immer mehr mit Android vertraut gemacht haben, war auch das Modell kein Problem mehr, sondern eher ein Vorteil, da es sehr logisch und durchdacht ist. Eine weitere Schwierigkeit, die während der Entwicklung aufgetreten ist, ist, dass es so viele verschiedene Android-Versionen und Geräte gibt. Da wir unsere App für so viele Versionen wie möglichen entwickeln wollten, kamen deshalb auch mal das ein oder andere Problem auf, wie z. B. dass manche Libraries oder Frameworks erst ab bestimmten Versionen verfügbar sind oder die vielen verschiedenen Geräte alle unterschiedlichen Seiten- und Größenverhältnisse haben.
Die Vorteile von Android überwiegen aber auf jeden Fall, vor allem, wenn man sich damit tiefer beschäftigt und eingearbeitet hat. Einer der größten Vorteile ist die sehr gute Dokumentation von Android durch die das Einlesen in die Möglichkeiten und Funktionen recht leicht ist. Auch die weltweite Verbreitung und Beliebtheit von Android ist hier ein Vorteil, da es sehr viele Entwickler gibt und so jedes Problem schon eimal aufgetreten ist und daher auch zu den allermeisten auch Lösungen oder Workarounds bekannt sind. Außerdem wird Android stetig Weiterentwickelt, weshalb immer mehr möglich ist und der Umgang mit bestimmten Funktionen, wie z. B. der Zugriff auf Gerätefunktionen wie Kamera oder Galerie immer leichter wird. Weitere Vorteile für uns sind die Vertrautheit mit Java und die Einfachheit von Android Studio.

2.3 Frameworks

Frameworks sind eine Art Gerüst oder Rahmen, die eingesetzt werden um das Programmieren zu vereinfachen und die geschriebenen Zeilen zu verringern.
Wir haben in unserer App drei Frameworks als Hilfen genutzt:

- Picasso: Erlaubt einfaches Handling (z. B. Größentransformationen) von Bildern in oftmals einer Zeile
- MPAndroidCharts: Ermöglicht die Erstellung von Diagrammen (in unserem Fall Tortendiagramme für die Statistiken)
- com.github.clans:fab: Ein fancy Menü, das schön ein- und ausgeklappt werden kann

3

Anforderungsanalyse

3.1 Funktionale Anforderungen

3.2 Nicht Funktionale Anforderungen

4

Konzept und Entwurf

5

Implementierung

blablabla

5.1 Implementierungsdetails

- keine Ahnung was da am besten bei uns erklärt werden kann - Algorithmen: möglichwerweise die "KIöder die Punkteberechnung - vielleicht Galerie-Darstellung durch Grid-Layout-Adapter?

5.2 Architektur

- ein paar Einleitungssätze - Datenmodell aus Präsentation kopieren und erlüutern, was besonders ist (auf Speicherung mit JSON eingehen) - Klassen-/Activity-Modell aus Präsentation kopieren - Download- und Upload-Verlauf-Modell (muss noch erstellt werden) - vielleicht ein paar Sätze zu allgemeinem Vorgehen und Aufteilung??

5.3 Besonderheiten

- Deckcreator und -editor vorstellen - Up- und Download von Decks - ...

5.4 Schwierigkeiten während der Implementierung

Wie in Präsentation

6

Anforderungsabgleich

6.1 Funktionale Anforderungen

6.2 Nicht Funktionale Anforderungen

7

Zusammenfassung und Ausblick

- Zusammenfassung: gute App geworden und wir haben viel gelernt ... - Ausblick was die App angeht: App eigentlich fertig aber könnte an manchen Stellen noch optimiert werden (Design, weitere Funktionen,...) - Ausblick was das Team angeht: erlerntes Wissen in das Entwickeln neuer Apps umsetzen

A

Quelltexte

In diesem Anhang sind einige wichtige Quelltexte aufgeführt.

```
1 public class Hello {
2     public static void main(String[] args) {
3         System.out.println("Hello World");
4     }
5 }
```

Listing A.1: Zeilencode

Abbildungsverzeichnis