

Mobile Application Lab

Dokumentation

Max Landthaler, Manuel Konstantin Hinke

07.03.2018

1 Einleitung

1.1 Motivation

Im Rahmen des Anwendungsfaches Mobile Application Lab wurde eine Quartett App geschrieben. Ziel des ganzen war es einen grundlegenden Einstieg in die Android Programmierung zu finden. Da ein Spiel viele verschiedene Bereiche abdeckt eignet es sich ideal für den Lernprozess. Übergeordnete Motivation war der Lernprozess wofür sich dieses Szenario sehr gut eignet.

1.2 Ziel

Das hiermit erklärte Ziel dieses Projekts ist die Entwicklung einer Quarett-App fortwährend auch Unicorn-Quartett genannt. Es sollen gängige und Vorgehensweisen und Best Practices erarbeitet werden. Dabei ist auch Ziel herauszufinden welche Schwierigkeit es gibt beim Implementieren für Android Betriebssysteme unterschiedlicher Versionen zu entwickeln.

1.3 Aufbau

Die folgende Dokumentation beschreibt den Enwicklungs und Implementierungsablauf. Dabei werden zuerst Grundlegende Rahmen definiert und eine Anforderungsanalyse erstellt, die sowohl funktionale als auch nicht funktionale Anforderungen erfüllt. Auch das erdachte Design und Struktur Konzept wird beschrieben um auch eventuell vergleichen zu können wo sich das Endresultat mit dem tatsächlich erreichten unterscheidet. Es

folgen dann ausgewählte Implementierungsdetails und abschließend ein Abgleich der Anforderungen.

2 Grundlagen

2.1 Quartettspiel

Beim klassischen Quartett geht das darum, möglichst viele Quartette (Ansammlung von 4 Karten gleicher Kategorie) zu erspielen. Dabei werden die Werte der Kontrahenten verglichen und der höhere Wert gewinnt. Der Gewinner bekommt die unterlegenden Karten. Das Unicorn-Quartet spielt im Standardmodus eine Abwandlung des klassischen Quartetts, das sich Supertrumpf nennt bzw. im Umgangssprachlichen als Autoquartett bezeichnet wird. Es folgen die genauen Beschreibungen der Spielelemente und Spielmodi.

2.1.1 Quartettkarte



Abbildung 1: Die Geniesserin aus dem Bettssport Deck

In Abb.1 wird eine Quartettkarte im Unicorn-Quartett visualisiert. Ganz oben wird der aktuelle Spielstand angezeigt und wer aktuell am Zug ist. Dann folgt ein Bild, auf dem das entsprechende Objekt der Karte zu sehen ist, in diesem Fall ein Auto. Unten werden die zur Auswahl stehenden Werte angezeigt.

2.1.2 Spielverlauf

An einem Spiel nehmen der Spieler und eine Künstliche Intelligenz als Kontrahent teil. Der menschliche Spieler legt ein Quartett und den Spielmodus fest. Hierbei kann sich zwischen einem normalen Spiel und dem sog. Unicorn Modus entscheiden. Zu Spielbeginn werden die Karten des Quartetts gemischt und gleichmäßig an die beiden Kontrahenten verteilt. Der menschliche Spieler ist immer zuerst an der Reihe und wählt einen Wert von seiner aktuellen Karte aus, mit dem er seinen Kontrahenten stechen möchte. Nachdem er seinen Wert bestätigt hat, werden sein Wert und der des Kontrahenten verglichen. Wer den besseren Wert hatte, bekommt die Karte des Verlierers. Anschließend ist der andere Spieler an der Reihe, d.h. es wird unabhängig vom Ausgang der vorherigen Runde immer abgewechselt. Gewinner des Spiels ist ganz klassisch der Spieler, der am Ende alle Karten hat.

Um etwas Abwechslung in den normalen Quartett Spielfluss zu schaffen, wurde der Unicorn Modus entwickelt.

2.1.3 Spielmodi

Normal: Der klassische Modus. Hier wird abwechselnd gespielt. Gewonnen hat derjenige, der am Ende alle Karten hat.

Unicorn: Der überraschende Modus. Auch hier wird abwechselnd gespielt. Jedoch können beide Kontrahenten ab einer Siegesserie von 3 hintereinander gewonnen Runden, sofern sie selbst an der Reihe sind, besondere, zufällig generierte Aktionen ausführen, die zu ihrem direkten Vorteil sind. Außerdem werden zufällig generierte Events, die beide Spieler positiv oder negativ beeinflussen, in das Spiel gestreut, ohne dass die Spieler darauf Einfluss nehmen können. Gewonnen hat der Spieler, der alle Karten hat oder die richtige Aktion ausgespielt hat.

2.2 Mobile Anwendungen

Ziel dieses Projektes war es, sich mit der Entwicklung von mobilen Anwendungen auf einer beliebigen Plattform zu befassen und erste Erfahrungen zu sammeln.

Da beide Entwickler ein Android Endgerät besitzen und auch Interesse an dieser Umgebung haben, war sofort Android als Plattform gesetzt.

Weitere Vorteile bei der Entwicklung von Android Anwendungen sind die große Verbreitung von Android Endgeräten und die Verwandtheit zur normalen Java Entwicklung.

Im Hinblick auf mobile Anwendungen fallen besonders die beschränkten Ressourcen wie begrenzte Akkukapazität, Arbeitsspeicher, Rechenleistung und Festplattenspeicher ins Gewicht. Im Vergleich zu normalen PCs oder Servern stehen auf einem mobilen Gerät viel weniger Ressourcen zur Verfügung, die zudem von vielen Anwendungen sowie dem System gleichzeitig verwendet werden.

2.3 Libraries und Frameworks

Um einige Funktionen der Applikation nicht selbst implementieren zu müssen wurden folgende zusätzliche Programmbibliotheken und Frameworks verwendet:

Volley, <https://github.com/google/volley>

Volley ist eine Bibliothek, die es sich zur Aufgabe gemacht hat, Netzwerkoperationen sehr einfach und effektiv durchzuführen. Mithilfe von Volley wurde das Downloaden von neuen Decks und Karten realisiert.

CircleImageView, <https://github.com/hdodenhof/CircleImageView>

Hiermit wird eine runde ImageView eingeführt, die für die Abbildung des Profil Bildes verwendet wurde.

Realm, <https://github.com/realm/realm-java>

Realm ist eine objektorientierte Datenbank, die sich gerade für Android eignet, um einfach und ohne echte Datenbankbefehle tatsächliche Objekte in einer Datenbank zu verwalten, was die Behandlung der einzelnen Spielstatistiken und Decks sowie deren Karten, Attribute und Bilder sehr einfach gestaltet.

3 Anforderungsanalyse

3.1 Funktionale Anforderungen

Ein wichtiges Thema im Vorfeld der Implementierung war es die Anforderungen an das Unicorn Quartett klar zu definieren. Im folgenden werden die funktionalen und nicht funktionalen Anforderungen aufgelistet, die vor der Entwicklung der Applikation formuliert wurden.

FA1 Ladebildschirm

Beim Öffnen der App erscheint ein Ladebildschirm, in dem das Logo der App dargestellt wird. Zudem befindet sieht der Nutzer eine Fortschrittsanzeige mit Statusmeldungen. Anschließend wird der Nutzer automatisch auf den Startbildschirm weitergeleitet.

FA1-0 Nutzer erstellen

Wenn der Nutzer zum ersten Mal die App startet, wird er dazu aufgefordert, einen Benutzer zu erstellen. Dies muss zu beim Start der App getätigt werden. Zur Erstellung des Nutzer wird ein Name und ein Profilbild welches wahlweise mit der Kamera aufgenommen werden kann oder aus der Galerie ausgewählt werden kann, benötigt. Über den Namen kann der Nutzer von anderen Nutzern identifiziert werden.

FA1-1 Startbildschirm

Auf dem Startbildschirm befinden sich 5 Elemente: Über Buttons mit entsprechenden Icons gelangt der Nutzer zu den folgenden Ansichten: - Spielen - Kartenverwaltung/Kartengallerie - Freunde - Rangliste - Profil

FA2-1 Spiel erstellen — Offline Spiel

Auf der Spielansicht kann der Nutzer zwischen zwei Optionen wählen: - Offline Spiel gegen die „KI“ - Online Spiel gegen Freunde Nach der Auswahl gelangt der Spieler auf die Ansicht, in der er sich zwischen den verschiedenen entscheiden kann.

FA2-2 Online Spiel

Wählt der Spieler die Option „online“ aus, kann er entweder gegen einen zufällig ausgewählten Online-Spieler spielen oder er wählt einen Spieler aus seiner Freundesliste aus.

FA2-3 Spielmodi

Der Nutzer kann zwischen zwischen zwei Spielmodi auswählen:

- Standard (Stechen)
- Unicorn Mode (Stechen, aber mit globalen Events und Sonderkarten)

FA2-4 Spiel

Im eigentlichen Spiel sieht der Spieler je Runde genau eine Karte mit aktuellem Bild und den Werten der Karte. Der Nutzer wählt einen Wert aus, der ihm am vielversprechendsten erscheint. Dann erfolgt ein Wechsel der Anzeige von der einzelnen Karte des Spielers zu einer Gegenüberstellung der beiden Karten der gegeneinander antretenden Spieler. Der Wert des Siegers wird grün markiert, der des Verlierers rot. Der aktuelle Spielstand wird eingeblendet und die nächste Runde beginnt. Dabei wird sichtbar visualisiert wer gerade am Zug ist.

FA3-1 Kartenverwaltung

In der Kartenverwaltung wird dem Spieler eine Übersicht aller möglichen Decks angezeigt. Vorhandene Decks werden in Farbe abgebildet, Decks, die erst mit der Zeit erworben werden können, sind zu Beginn ausgegraut. Klickt der Nutzer auf ein Deck, kann er sich die einzelnen Karten darin ansehen.

FA4-1 Bestenliste

In der Bestenliste werden die Top 5 Online Spieler angezeigt.

FA5-1 Freunde

In der Freundesübersicht kann der Nutzer entweder einen Freund über seinen Namen hinzufügen oder kann die Liste seiner Freunde ansehen, wenn diese die Freundschaftsanfragen angenommen haben.

FA6-1 Spieler Profil

Im Spielerprofil kann der Nutzer seinen Account mit Google verknüpfen, um seinen Fortschritt auch online zu speichern und um online spielen zu können. Außerdem kann er hier seine Spielhistorie einsehen.

3.2 Nicht Funktionale Anforderungen

NFA1 Aktuelle Version der Betriebssysteme

Die Oberfläche der App sollte sich an aktuelle Designrichtlinien orientieren und auf den gängigsten Versionen lauffähig sein.

NFA2 Usability Ziele

Die App sollte einfach und verständlich aufgebaut werden sodass sich der Nutzer intuitiv in den Menüs zurechtfinden kann.

NFA3 Feedback

Die App sollte dem Nutzer unter bestimmten Bedingungen ein Feedback geben. Der Nutzer darf nie im Unklaren darüber sein was die App gerade macht.

NFA4 Selbsterklärbarkeit Die App sollte selbsterklärend aufgebaut sein. Der Nutzer sollte keine Probleme beim Verständnis der Funktionen oder Icons haben und ohne Hilfe das System bedienen können. Die App sollte ohne eine Anleitung bedienbar sein.

NFA5 Benutzerfreundlichkeit

Die App sollte benutzerfreundlich aufgebaut sein, dass heißt der User soll einen positiven Eindruck von der App bekommen und sie gerne verwenden.

NFA6 Verfügbarkeit

Die App sollte in den üblichen Smartphone Auflösungen darstellbar sein und sich auch an die Größenänderungen des Displays anpassen.

NFA7 Robustheit

Die App sollte gegenüber Fehleingaben robust sein.

NFA8 Reaktionszeit

Die App soll innerhalb so kurzer Zeit reagieren, dass der Nutzer keine nennenswerte Wartezeiten abwarten muss, ohne zu wissen, was vor sich geht.

NFA9 Statusrückmeldung

Wenn die App nicht innerhalb angenehmer Reaktionszeiten reagieren kann, soll der Nutzer über eine aussagekräftige Anzeige über den aktuellen Zustand der App informiert werden.

4 Konzept und Entwurf

4.1 Mockups

Im Vorfeld der Implementierung musste natürlich auch ein Entwurfskonzept erarbeitet werden. Dieses Entwurfskonzept sollte die grundlegende Architektur der Applikation beschreiben und die einzelnen Screens in einer schlichten und funktionalen Art vorbereiten. Wir haben bei der Erstellung dieser Screens auf eine softwareunterstützte Lösung verzichtet und haben diese per Hand erstellt.

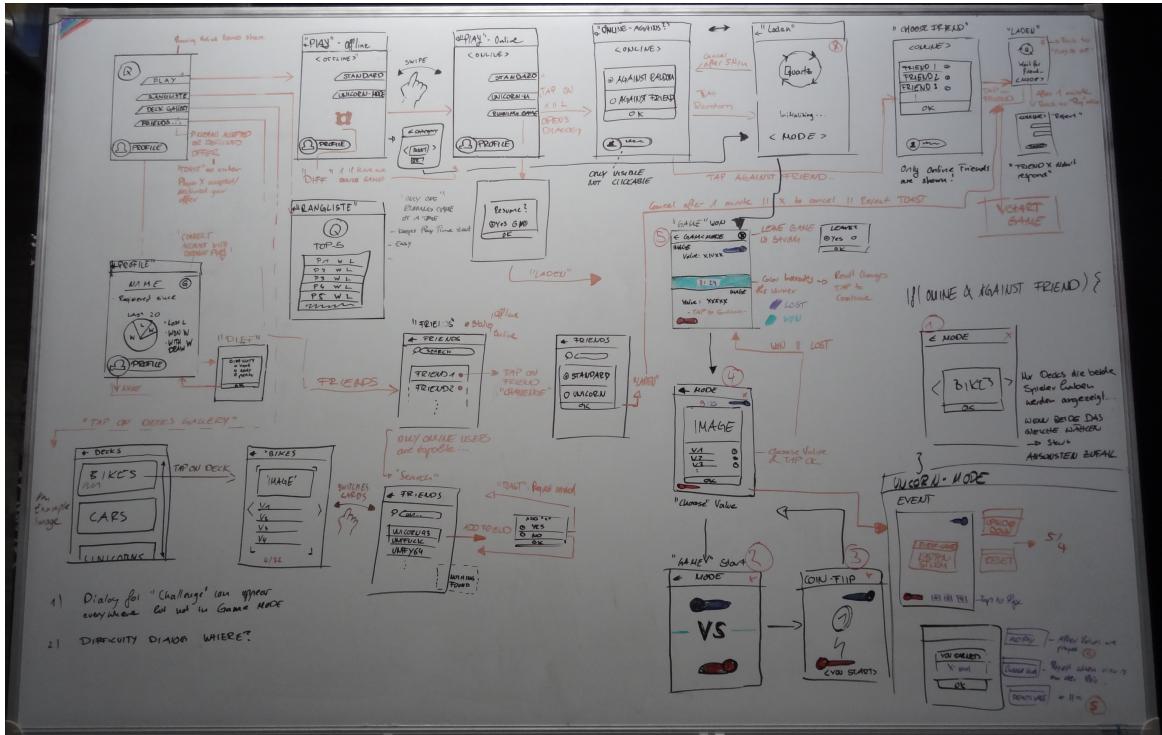


Abbildung 2: Komplettansicht-Prozessablauf

Darstellung der gesamten Möglichkeiten beziehungsweise Prozesse die in der App realisiert werden sollten. Das Profil ist von allen Screens aus erreichbar es sei denn man befindet sich in einem Spielprozess.

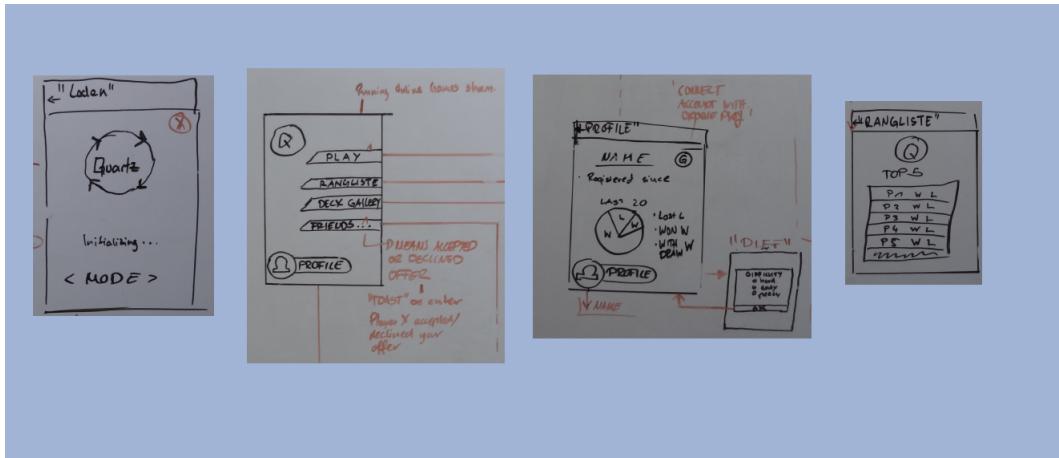


Abbildung 3: Erstellung des Benutzers und bearbeiten der Daten

Abbildung 3 skizziert den Prozess, welchen der Benutzer vollführen muss, um einen Nutzer zu erstellen und wie er eingegebene Daten später wieder ändern kann.

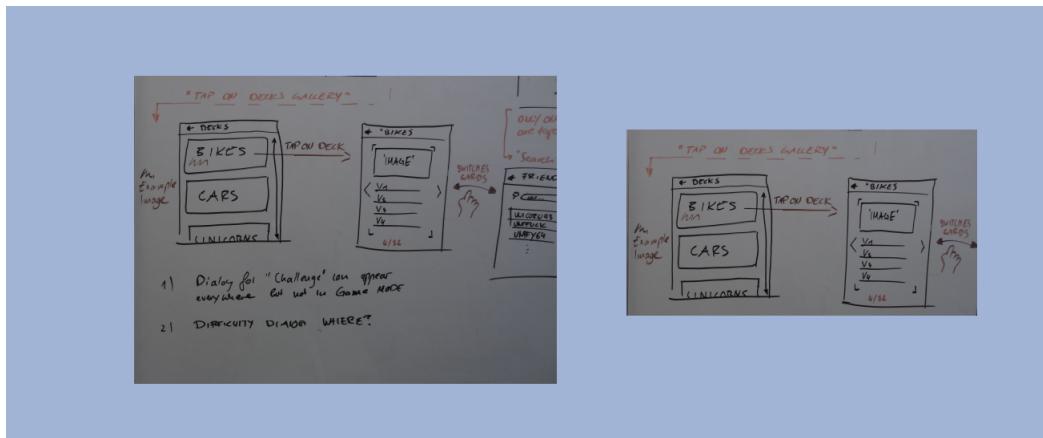


Abbildung 4: Kartenansicht/Kartengallerie

In der Abbildung 4 wählt der Nutzer eines seiner Decks aus und kann sich dessen Karten ansehen.

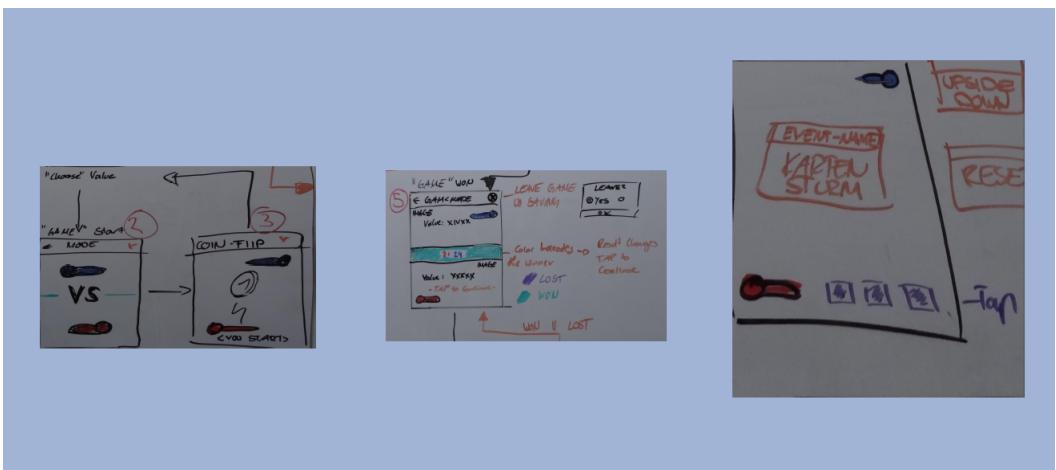


Abbildung 5: Spielablauf

Diese Screens zeigen den grundlegenden Ablauf eines Spiel. Diese sind zwar unterschiedlich je nach Modi aber zum Zeitpunkt der Mockups ging es nur um ein generelles Darstellen des Prozesses. 5 zeigt.

5 Implementierung

5.1 Ausgewählte Implementierungsdetails

5.1.1 Startseite — Profil

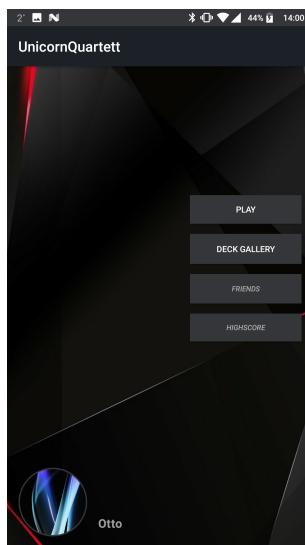


Abbildung 6: Startseite



Abbildung 7: Profil

Wenn der Nutzer das erste Mal die Anwendung startet, wird er aufgefordert, einen Benutzernamen einzugeben und ein Profilbild anzugeben. Zudem muss er sich für ein Darstellungsthema entscheiden. Dieses kann er entweder mit der Kamera des Smartphones aufnehmen oder ein bereits erstelltes Bild aus der Galerie verwenden. Das Profilbild wird nun in allen Übersichtsscreens am unterem linken Bildschirmrand angezeigt. Beim erstmaligem Starten der Anwendung wird somit ein User Objekt angelegt, welches anschließend in der Realm Datenbank persitiert wird.

Klickt der Nutzer auf sein Profilbild, kommt er zur Profilansicht. Dort kann er seinen Nutzernamen, das Profilbild und das Darstellungsthema ändern. Zudem kann der Nutzer hier allerdings auch die Schwierigkeitsstufe der künstlichen Intelligenz auswählen. Er erhält weiterhin eine individuelle Übersicht einer internen Auswertung über seine gesamte Spielhistorie, welche abhängig von dem Verhältnis SSieg - Niederlage unterschiedliche Rückmeldungen an den Nutzer bietet. All diese Konfigurationen werden in der Datenbank im User Eintrag hinterlegt und ggf. aktualisiert. Die Spielhistorie wird anhang von einer in der Datenbank hinterlegten Liste an Games analysiert.

5.1.2 Deckansicht

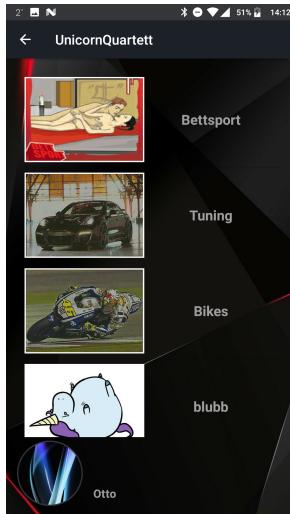


Abbildung 8: Gallerie

In der Deckansicht kann man alle zum Download stehenden bzw. bereits heruntergeladenen Decks einsehen. Beim Start der Activity wird überprüft, welche Decks auf dem Server zum Download verfügbar sind und welche davon bereits auf dem Endgerät heruntergeladen wurden. Steht ein Deck zum Download bereit, kann auf den entsprechenden Platzhalter geklickt werden und der Download beginnt. Sämtliche Netzwerkaktionen wurden über Volley realisiert. Zudem kann ein Deck nur bei bestehender WLAN-Verbindung heruntergeladen werden. Sollte dies nicht der Fall sein, wird der Nutzer darüber informiert. Dabei werden nacheinander alle nötigen Informationen über das Deck, sein Attributschema und die einzelnen Karten heruntergeladen und in temporäre Objekte zum Bauen von den tatsächlich in der Datenbank hinterlegten für das Spiel notwendigen Objekten zusammengefasst. Nach dem tatsächlichen Download werden aus diesen temporären BuildObjekten in einem weiteren Prozess die tatsächlichen Objekte für die Datenbank erstellt. Ist der Vorgang erfolgreich abgeschlossen, werden die temporären Dateien gelöscht und die Deckansicht refreshet. Während dem Entwicklungsprozess traten auf manchen Endgeräten immer wieder *OutOfMemoryErrors* auf. Um diese zu verhindern, musste jedes Bild über einen Algorithmus relativ zu seiner ursprünglichen Auflösung heruntergerendert werden. Durch einen Klick auf ein heruntergeladenes Deck

gelangt man in die Kartenansicht des Decks, in dem Karten des Decks inkl. ihrer Werte angezeigt werden.

5.1.3 Das Spiel



Abbildung 9: Das Spiel

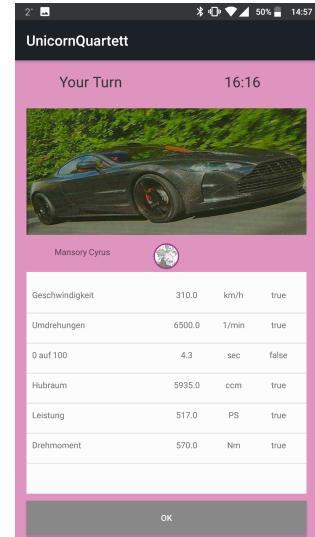


Abbildung 10: Unicorn Modus

Der Nutzer kann zwischen zwei verschiedenen Spielmodi auswählen: Dem "normalen Standard Modus und den speziellen Unicorn Modus. Beide verfolgen grundlegend das gleiche Prinzip: Abwechselnd wählen die Kontrahenten einen Wert von ihrer Karte aus. Der Gewinner, also der Spieler mit dem besseren Wert, bekommt die Karte des Verlierers. Dabei wird nach jeder Runde der aktuelle Zwischenstand, die jeweiligen Kartenstapel sowie der StichKartenStapel, der angelegt wird, sobald beide Spieler den gleichen Wert haben, in einem Game Objekt in der Datenbank hinterlegt. Das heißt, dass jeder Zeit das aktuelle Spiel unterbrochen werden kann und erst später, auch nach dem Beenden der Anwendung, wieder an der gleichen Stelle fortgesetzt werden kann. Zudem hat der Nutzer die Möglichkeit, ein angefangenes Spiel zu löschen und neu oder mit einem anderem Deck anzufangen.

Im Unicorn Modus hingegen beeinflussen zufällig generierte Events und vom Spieler eingesetzte Boni das Spielgeschehen. Das grundlegende Spielprinzip bleibt jedoch erhalten.

Die Implementierung der künstlichen Intelligenz hat sich sehr einfach gestaltet. Je nach

Schwierigkeitsgrad entscheidet die KI zufällig oder aus einem Bereich der prozentual anhand dem Durchschnitt der Wertekategorien im Deck berechneten Werte. Die KI kennt dabei wie ein menschlicher Spieler die Karten des Gegners nicht und kann somit nicht unfairer Weise strategische Entscheidungen bzgl. der Werte seines Gegners treffen.

5.2 Architektur

5.2.1 Datenmodell

Das Datenmodell dieser Anwendung basiert auf REALM. Realm ist ein Datenbank Management System, welches insbesondere für mobile Plattformen wie Android gedacht ist. Realm bietet sich vor allem für objektorientierte Anwendungsansätze wie Android an, da mithilfe von Realm sehr einfach Objekte abgebildet werden und diese ohne Mehraufwand seitens des Programmierers in die Datenbank gespeichert und wieder ausgelesen werden können. Realm bietet die Möglichkeit, Operationen auf einer Datenbank auszuführen, ohne klassische SQL Befehle einsetzen zu müssen. Diese werden implizit von Realm gehandhabt.

5.2.2 Klassenstruktur

Bei der Implementierung wurde versucht, sich an das MVC Pattern zu halten. Dies gelang jedoch stellenweise nicht, weil aufgrund der fehlenden Erfahrung noch eine zu hohe Kopplung zwischen Views und Moduel in den Activities vorhanden ist.