

Smartphone-App zum Spielen von analogen Kartenspielen mit einem französischen Blatt

Zeitraum: 12.09.2019 – 14.11.2019

Teammitglieder:

Matthias Brehmer

Sandra Gall

Max Naujokat

Doreen Oesterle

Tim Wißler

Tobias Krämer

Inhaltsverzeichnis

Beschreibung des Projektes	3
Genutzte Tools	4
Android Studio	4
draw.io	4
GitHub	5
Scrum	5
Epic	7
Sprint 1: 12.09.2019 – 19.09.2019	8
Sprint 2: 19.09.2019 – 26.09.2019	10
Sprint 3: 26.09.2019 – 10.10.2019	11
Sprint 4: 10.10.2019 – 17.10.2019	12
Sprint 5: 17.10.2019 – 24.10.2019	15
Sprint 6: 24.10.2019 – 31.10.2019	16
Sprint 7: 31.10.2019 – 07.11.2019	17
Sprint 8: 07.11.2019 – 14.11.2019	18
Fazit zu dem Projekt	19

Beschreibung des Projektes

In unserem Projekt geht es um die Entwicklung einer App, auf der man, falls man kein “reales” Kartendeck haben sollte, digital zusammen Kartenspiele mit einem französischen Blatt spielen kann. Die Apps kommunizieren mit einem Server, so dass jeder auf seinem Smartphone seine Hand, sowie Karten, die in der Mitte liegen, sehen kann. Man kann die Karten jedoch nicht nur sehen, sondern auch z.B. Karten ziehen oder diese ablegen.

In die App ist kein spezielles Spiel, wie z.B. Mau-Mau einprogrammiert, sondern sie gibt die Möglichkeit vor Spielbeginn bestimmte Parameter zu definieren, wie z.B. die Anzahl und Größe der Stapel in der Mitte oder die Anzahl der Karten auf der Hand der Spieler am Anfang des Spiels. Die App liefert also nur eine “digitale Umgebung” zum Spielen, die (zeitliche) Logik für das jeweilige Spiel muss immer noch von den Spielern umgesetzt werden.

Genutzte Tools

- Android Studio
- draw.io
- GitHub
- Scrum

Android Studio

Als Tool zur Erstellung der App wurde Android Studio gewählt. Es ist eine freie Integrierte Entwicklungsumgebung von Google. Das Programm basiert auf IntelliJ IDEA.

Neben den Funktionen, die bereits in IntelliJ IDEA implementiert sind, stehen die folgenden zur Verfügung:

Unterstützung für die Entwicklung von Android, Android Wear und Android TV Apps.

- Android Studio verwendet ein Build-Management-Automatisierungs-Tool, das auf Gradle basiert. Das System gibt dem Entwickler die Möglichkeit, für verschiedene Gerätetypen, wie z. B. Tablets, optimierte Versionen der App zu erstellen.
- Theme-Editor
- Android Lint
- Es ist möglich, Google-Dienste wie Google Cloud Messaging innerhalb der IDE zu konfigurieren und direkt auf die App anzuwenden.
- Unterstützung für ProGuard und automatische App-Signierung

Der Quelltext von Android Studio ist frei verfügbar.

draw.io

Mit der Desktop-App „Draw.io“ kann man Diagramme, Organigramme und Mindmaps sowie Schaltkreise, Grundrisse, Webseiten-Layouts, Businessmodelle und Übersichten zur Dokumentation bei der Softwareentwicklung erstellen. Das Programm bietet Vorlagen aus Kategorien wie Business, Netzwerk, Technik und Layout sowie UML-Diagramme, Venn-Diagramme und verschiedene Infrastrukturen für den „Google Cloud Platform“-Service. Neben Bearbeitungsmöglichkeiten für Grafiken und Texte verfügt „Draw.io“ über eine Bilddatenbank mit einer großen Auswahl an Formen für diverse Einsatzzwecke. „Draw.io“ kann mit externen Plugins um neue Funktionen erweitert werden, um z.B. Texte aus Diagrammen zu extrahieren oder sensible Daten zu verfremden. Wenn man bei einem Cloud-Dienst wie Dropbox, OneDrive oder Google Drive angemeldet ist, können die Projekte mithilfe der Cloud-Dienste gespeichert jederzeit synchronisiert werden. Das Ergebnis kann als Bilddatei in den Formaten PNG, JPEG und SVG oder als PDF-Dokument exportiert werden. Zudem besteht die Möglichkeit, das Diagramm über eine URL freizugeben, auf einer Webseite einzubetten oder in die Online-Projektmanagementsoftware Trello zu importieren.

GitHub

GitHub ist ein Onlinedienst, der Software-Entwicklungsprojekte auf seinen Servern bereitstellt. Im Gegensatz zu anderen Dienstleistern zur Verwaltung quelloffener Software, wie SourceForge, steht auf GitHub nicht das Projekt als Sammlung von Quellcode im Zentrum, sondern der Nutzer mit seinen Quelltext-Datenbanken, den sogenannten Repositories (also Verzeichnissen, die mit Git verwaltet werden). Auch das Erstellen und Zusammenführen von Abspaltungen (engl. ‚forks‘) wird besonders propagiert. Die sog. Forks machen das Mitentwickeln bei fremden Projekten besonders einfach: Um dort einen Beitrag beizusteuern, wird das Repository zunächst abgespalten, dann werden die zu übernehmenden Änderungen hinzugefügt und dem Besitzer des Originals eine gestellt, die Änderungen zu übernehmen. Da alle Schritte auch über die Weboberfläche möglich sind, ist die Bedienung von GitHub im Vergleich zu anderen Entwicklerplattformen auch für Anfänger besonders einfach. Damit wird unter Berücksichtigung der Besonderheiten verteilter Versionskontrollsysteme ein soziales Netzwerk geschaffen, was sich auch in den aus „echten“ sozialen Netzwerken bekannten Funktionen „Beobachten“ oder „Folgen“ zeigt.

Sowohl das Erstellen öffentlich einsehbarer als auch privater, also nichtöffentlicher Repositories ist nach einer kostenlosen Registrierung möglich. Außerdem bietet GitHub größeren Unternehmen mit GitHub Enterprise an, eine eigene, abgetrennte GitHub-Installation zu liefern, um die Vorteile des sozialen Programmierens auch bei der unternehmensinternen Softwareentwicklung zu nutzen.

Scrum

Als Vorgehensmodell des Projekt- und Produktmanagements haben wir Scrum gewählt, welches insbesondere zur agilen Softwareentwicklung dient. Scrum kennt drei Rollen für direkt am Prozess Beteiligte: Product Owner (stellt fachliche Anforderungen und priorisiert sie), ScrumMaster (managt den Prozess und beseitigt Hindernisse) und Team (entwickelt das Produkt). Daneben gibt es als Beobachter und Ratgeber noch die Stakeholders.

Die Anforderungen (Requirements) werden in einer Liste (Product Backlog) gepflegt, erweitert und priorisiert. Das Product Backlog ist ständig im Fluss. Um ein sinnvolles Arbeiten zu ermöglichen, wird monatlich vom Team in Kooperation mit dem Product Owner ein definiertes Arbeitspaket dem oberen, höher priorisierten Ende des Product Backlogs entnommen und komplett in Funktionalität umgesetzt (inkl. Test und notwendiger Dokumentation). Dieses Arbeitspaket, das Increment, wird während der laufenden Iteration, des sog. Sprints, nicht durch Zusatzanforderungen modifiziert, um seine Fertigstellung nicht zu gefährden. Alle anderen Teile des Product Backlogs können vom Product Owner in Vorbereitung für den nachfolgenden Sprint verändert bzw. neu priorisiert werden.

Das Arbeitspaket wird in kleinere Arbeitspakete (Tasks) heruntergebrochen und mit jeweils zuständigem Bearbeiter und täglich aktualisiertem Restaufwand in einer weiteren Liste, dem Sprint Backlog, festgehalten. Während des Sprints arbeitet das Team konzentriert und ohne Störungen von außen daran, die Tasks aus dem Sprint Backlog in ein Increment of Potentially Shippable Functionality, also einen vollständig fertigen und potentiell produktiv einsetzbaren Anwendungsteil,

umzusetzen. Das Team gleicht sich in einem täglichen, streng auf 15 Minuten begrenzten Informations-Meeting, dem Daily Scrum Meeting, ab, damit jeder weiß, woran der andere zuletzt gearbeitet hat, was er als nächstes vor hat und welche Probleme es evtl. gibt.

Am Ende des Sprints präsentiert das Team dem Product Owner, den Stakeholders u.a. interessierten Teilnehmern in einem sog. Sprint Review Meeting live am System die implementierte Funktionalität. Halbfertiges oder gar Powerpoint-Folien sind während des Reviews verboten. Das Feedback der Zuseher und die neuen Anforderungen des Product Owners für den kommenden Sprint fließen dann wieder in das nächste Sprint Planning Meeting ein, und der Prozess beginnt von neuem.

Der ScrumMaster sorgt während des gesamten Prozesses dafür, dass Regeln eingehalten werden und der Status aller Tasks im Sprint Backlog von den jeweils zuständigen Team-Mitgliedern täglich aktualisiert wird. Er macht den Projektfortschritt transparent durch einen geeigneten Reporting-Mechanismus: die Veröffentlichung sog. Burndown Charts, welche den Fortschritt für den aktuellen Sprint bzw. für das gesamte Projekt jeweils in Form einer Kurve visualisieren. Eingezeichnete Trendlinien erlauben es, mögliche Probleme und Verzögerungen einfach (und rechtzeitig!) zu erkennen.

Im Kern basiert Scrum also auf einer inkrementellen Vorgehensweise, der Organisation von Entwicklungsabschnitten und Meetings in vordefinierten Zeitabschnitten (Time-Boxes) und der Erkenntnis, dass ein funktionierendes Produkt wichtiger ist als eine dreihundertseitige Spezifikation.

Epic

Was muss ich tun, um ein Spiel zu generieren?

Als Spieler möchte ich verschiedene Parameter eingeben können, um mein eigenes Kartendeck zu erstellen.

Als Spieler möchte ich mein erstelltes Spiel speichern können.

Als Spieler möchte ich gängige Spiele nicht erst konfigurieren müssen.

Als ProductOwner möchte ich eine einfache Bedienung der Spielgenerierung haben, damit viele die App nutzen.

Wie kann ich einem Spiel beitreten?

Als Spieler möchte ich mich anderen Spielern verbinden können.

Als Spieler möchte ich andere Spieler zu meinem Spiel einladen können.

Was für Einstellungen kann ich vornehmen?

Als Spieler möchte ich meinen Benutzernamen ändern können.

Als Spieler möchte ich meine Hintergrundmusik regeln können.

Brauche ich einen Nutzernamen?

Als Entwickler möchte ich einen Benutzernamen der Spieler, damit die Verbindung der einzelnen Spieler leichter hergestellt werden kann.

Als ProductOwner möchte ich einen Benutzernamen, um eine Statistik über das Spiel erstellen zu können.

Kann ich auch mit anderen Spielern spielen?

Als Spieler möchte ich mit meinen Freunden spielen können.

Als ProductOwner möchte ich eine Vernetzung von Spielern ermöglichen, damit das Spiel gemeinsam gespielt werden kann.

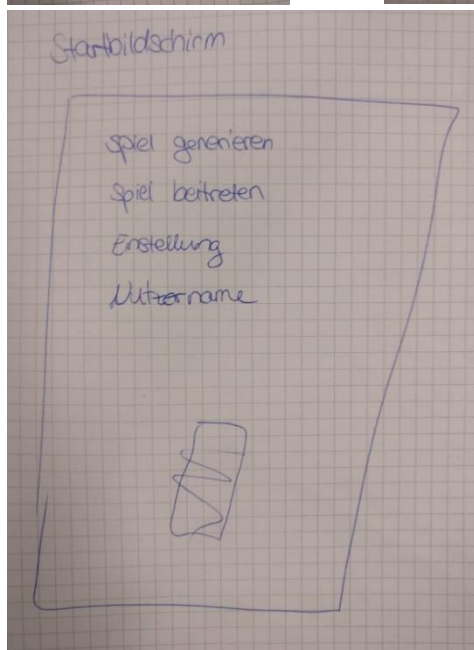
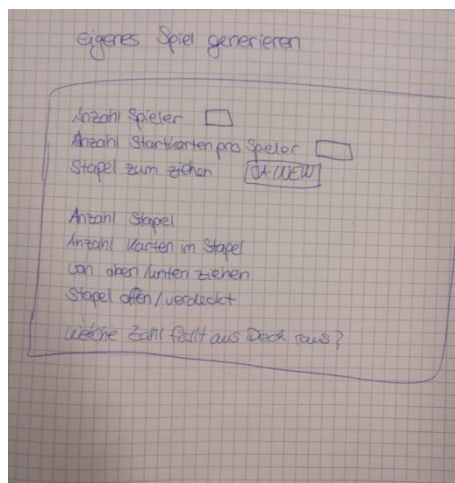
Sprint 1: 12.09.2019 – 19.09.2019

Planung:

- Wie soll die App allgemein funktionieren?
 - Verbindung über den Laptop
 - Server ist auf einem Smartphone gehostet – Spielleiter
 - Programmiersprache: Java
- Design planen

Durchführung:

Bilder von Design gezeichnet. Grundfunktionen der App bestimmt.



Review:

Zu dem Design wurden Mockups erstellt mit denen nun das Layout erstellt werden kann. Es ist nun klar welche Grundfunktionen die App haben soll und wie die App aufgebaut sein soll.

Sprint 2: 19.09.2019 – 26.09.2019

Planung:

- Der Server muss aufgebaut werden
- Android Studio muss ausprobiert werden
- IP Adresse und Android?
- Namenfindung für die App

Durchführung:

- Installation von Android Studio
- Ausprobieren von Android Studio
- App Name: Zonk
- IP Adresse kann am Handy angezeigt werden

Review:

Grundfunktionen von Android Studio sind nun bekannt.

Zonk kann als Name nicht verwendet werden, da urheberrechtlich geschützt.

-> Namensänderung

Eine Mitarbeiterin hat gekündigt. Ein neuer Mitarbeiter (Tobias Krämer) wurde als Ersatz eingestellt.

Sprint 3: 26.09.2019 – 10.10.2019

Planung:

- Ausarbeitung des Zusammenhangs der Objektstruktur und der graphischen Oberfläche
- Verbindung zum Server herstellen
- Entwickeln eines Logos
- App – Name ändern

Durchführung:

- Ausarbeitung des Zusammenhangs der Objektstruktur und der graphischen Oberfläche – wurde nicht gemacht
- App Name:
Münzwurf hat entschieden, der neue App-Name lautet Schlonk!
- Logo sketch wurde entwickelt



- Verbindung zum Server wurde hergestellt.

Review:

Aufgaben im Sprint wurden nur teilweise erledigt. Die Ausarbeitung des Zusammenhangs der Objektstruktur und der graphischen Oberfläche ist nicht zufriedenstellend. Muss unbedingt erledigt werden.

Das Logo muss noch verfeinert werden, welches wir verwenden können.

Die Verbindung zum Server steht, diese muss jedoch noch getestet werden.

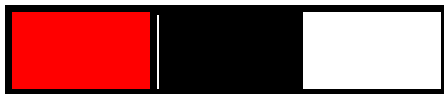
Sprint 4: 10.10.2019 – 17.10.2019

Planung:

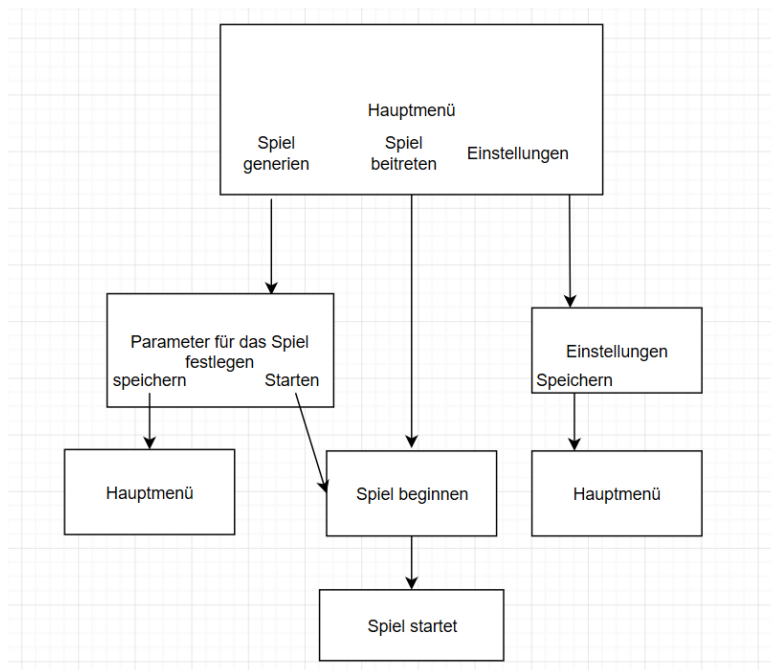
- Erstellen der Graphischen Oberfläche
- Ablaufplan der App-Seiten erstellen
- Farben der App festlegen
- Fertigstellung des Logos
- Ausarbeitung des Zusammenhangs der Objektstruktur und der graphischen Oberfläche
- Objektstruktur ausarbeiten

Durchführung:

- Graphische Oberfläche ist in Arbeit mit Android Studio
- Farbspektrum



- Ablaufplan der App

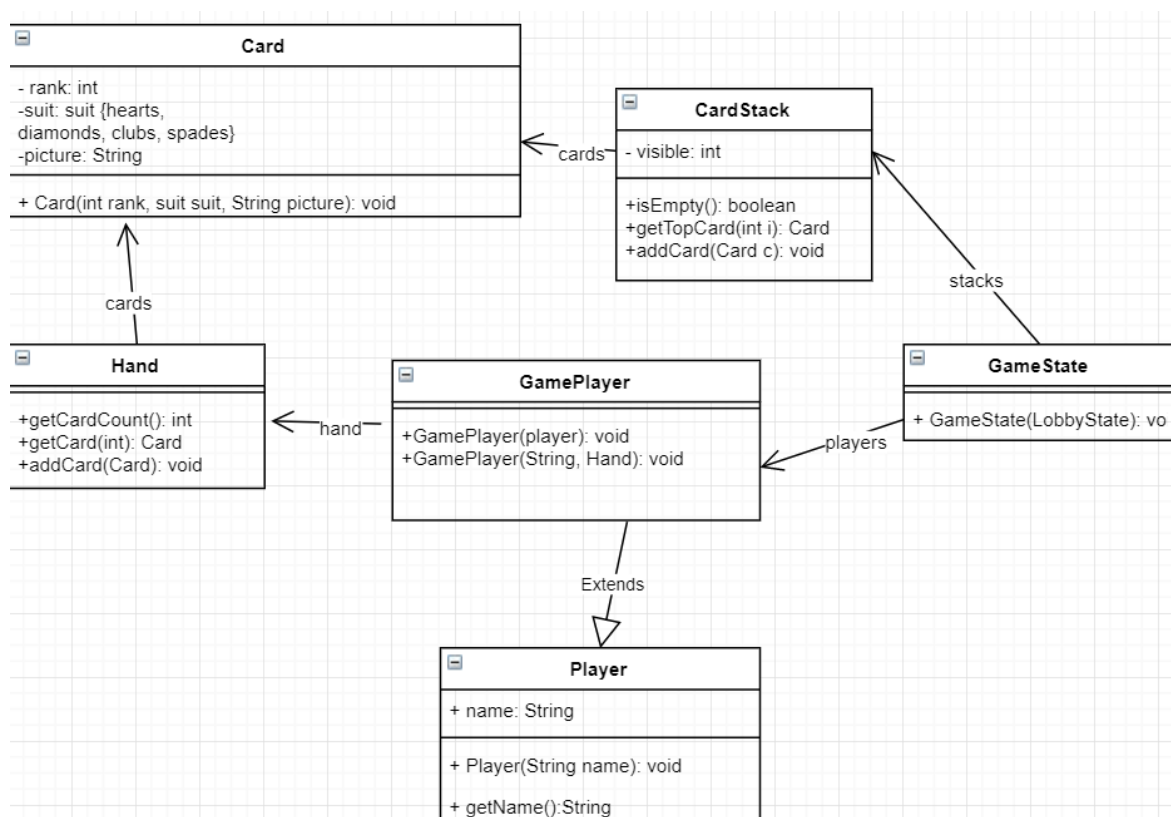


- Fertiges Logo



- Ausarbeitung des Zusammenhangs der Objektstruktur und der graphischen Oberfläche – konnte wegen Krankheit nicht ausgearbeitet werden
- Die Objektstruktur wurde ausgearbeitet, jedoch nicht implementiert

Klassendiagramm:



Review:

Wegen Krankheit sind wir nicht so weit gekommen wie wir wollten. Die Struktur unserer App ist erstellt – der Grundbaustein um implementieren zu können.

Sprint 5: 17.10.2019 – 24.10.2019

Planung:

- Implementierung der Objektstruktur
- Implementieren der App-Seiten um ins Spiel zu gelangen

Durchführung:

- Implementierung der Objektstruktur, die Klasse GameState, CardStack, GamePlayer wurden erstellt
- Implementieren der App-Seiten um ins Spiel zu gelangen – Fortschritt 10%

Review:

Absprache der einzelnen Unterthemen muss noch erfolgen. Verbindung, sodass alles zu einer einzelnen App kombiniert wird fehlt noch. Android Emulator bereitet Probleme, da er sich von einigen Teammitgliedern nicht installieren lässt, daher verzögert sich die Implementierung der App-Seiten. Die Aufgabe wurde einem anderen Teammitglied zugeteilt.

Sprint 6: 24.10.2019 – 31.10.2019

Planung:

- Graphische Oberfläche der einzelnen App-Seiten entwickeln
- Implementierung der Objektstruktur weiter machen
- Serververbindung implementieren

Durchführung:

- Die graphische Oberfläche wurde implementiert
- Serververbindung kann testweise hergestellt werden, volle Funktionalität nicht gegeben
- Implementierung der Objektstruktur (die Klassen Player, Hand und Card)

Review:

Das Design der graphischen Oberfläche war nicht optimal. Nach größeren Streitigkeiten hat man sich darauf verständigt das Layout der Oberfläche neu zu erstellen, die Grundfarben werden beibehalten.

Die Serververbindung kann aufgebaut werden, jedoch werden noch keine Spieldaten übertragen, da andere benötigte Teilaufgaben wie die eigentliche Spieleseite noch nicht implementiert sind. Diese hat nun höchste Priorität.

Die Objektstruktur wurde vollständig implementiert, nun muss die logische Funktionalität des Kartenspieles erstellt werden.

Sprint 7: 31.10.2019 – 07.11.2019

Planung:

- Graphische Oberfläche überarbeiten
- App-Seiten miteinander verbinden, sodass durch die Menüs navigiert werden kann
- Objektstruktur mit der graphischen Oberfläche verbinden
- Präsentation vorbereiten
- Serverbindung über graphische Oberfläche herstellbar machen
- Einbinden der Karten in die Graphische Oberfläche

Durchführung:

- Endgültiges Design erstellt
- Navigation durch APP wurde erstellt
- Karten können hinzugefügt und gelöscht werden
- Präsentation Layout erstellt
- Serververbindung kann hergestellt werden

Review:

Großer Fortschritt in diesem Sprint. Die Aufgaben wurden alle fristgerecht erledigt bis auf die Objektstruktur.

Sprint 8: 07.11.2019 – 14.11.2019

Planung:

- Fertigstellen der Objektstruktur in Verbindung mit der graphischen Oberfläche
- Serververbindung einbinden in Objektstruktur
- Musik hinzufügen als Unterhaltung während dem Spiel

Durchführung:

- Musik wurde hinzugefügt

Review:

In dem letzten Sprint ging es um die Verfeinerung und Fertigstellung des Programms. Zur besseren Unterhaltung wurde noch Musik hinzugefügt. Die anderen Aufgaben wurden aufgrund von Zeitmangel nicht umgesetzt.

Fazit zu dem Projekt

Es lässt sich feststellen, dass für unser Projekt Scrum als Vorgehensmodell nicht sehr vorteilhaft war. Die meisten Anforderungen an unsere Software waren bereits im Voraus bekannt und es gab keinen Kunden. Somit gab es keine Absprachen mit dem Kunden und auch keine neuen Anforderung, welche mit Scrum als agile Vorgehensweise gut bearbeitet werden könnten. Zudem war das Ziel, dass die App überhaupt funktioniert. Dies lässt sich nicht in auslieferbare Inkremente aufteilen. Bei fehlender Eigenmotivation ist es mit Scrum sehr schwer Fortschritte zu erreichen, was bei uns Anfangs der Fall war. Dieses Problem wurde noch verstärkt, da wir keinen ProductOwner hatten, der die Teammitglieder in Verantwortung gezogen hat und so das Projekt vorangetrieben hat. Gut war, dass wir unsere Fortschritte gut beobachten konnten und auch Aufgaben neu verteilen konnten, wenn es Probleme gab.

Zudem gibt es für eine erfolgreiche Teamarbeit einige Faktoren, welche uns Schwierigkeiten bereitet haben. So muss sich jeder an Absprachen halten und verlangte Anforderungen wie gewünscht erfüllen. Zusätzlich ist eine gute Kommunikation sehr wichtig. Bei uns war oft unklar, wer welche Aufgaben bearbeitet und vor allem wie diese bearbeitet werden. Dadurch hatten wir am Anfang viele einzelne Teile, die jedoch nicht zusammengepasst haben. Dieses Problem haben wir jedoch rechtzeitig erkannt und daraufhin unsere Kommunikation verbessert. Die einzelnen Teammitglieder haben ihre bereits bearbeiteten Aufgaben anpassen müssen, sodass alles zusammengefügt werden konnte. In unserem Fall wäre ein Verantwortlicher nötig gewesen, der am Anfang alles im Überblick hat und die Teammitglieder koordiniert. So hatten wir etwas Startschwierigkeiten, welche sich jedoch gelegt haben.

Große Probleme in unserem Projekt bereitete uns Android Studio und Android Emulator. Oft ist der Emulator nicht gestartet, oder zeigt Fehler an, wo keine sind. Android Studio hat sich bei einem Teammitglied nicht installieren lassen, bei einem anderen hat die Kommunikation mit GitHub nicht funktioniert. Die zwei Teammitglieder haben so mit Ihren eigenen Rechnern nichts tun können. Ein weiterer Störfaktor war eine längere Krankheit eines Teammitgliedes und ein Mitarbeiterwechsel. In beiden Fällen hat sich die Planung nach hinten verschoben, da Aufgaben neu verteilt werden mussten. Für uns ist die mangelnde Zeit auch ein Problem gewesen. Wir haben uns viel vorgenommen und dafür zu wenige Vorkenntnisse gehabt, was sehr viel Zeit beansprucht hat.

Die Grundzüge der App sind erstellt. Eine richtige Funktion ist jedoch noch nicht gegeben. Momentan kann kein Spiel gespielt werden, auch wenn einzelne Funktionen funktionieren. Wir haben am Anfang sehr viel Zeit verschwendet für Planung des Designs. Dabei haben wir die eigentliche Anforderung an unsere App, das Erstellen eines eigenen Kartendecks, aus dem Blick verloren. Nachdem wir jedoch unser Problem in der Zusammenarbeit erkannt hatten und unsere Kommunikation im Team besser wurde, sind noch sehr große Fortschritte erreicht worden. Unsere App hat ein sehr gutes Design mit eigenem Logo, es können sich zwei Emulator miteinander verbinden und den Spielstand synchronisieren, es können Karten auf einer Testseite bewegt werden und gelöscht bzw. hinzugefügt werden und allen benötigten App-Seiten, um in ein Spiel zu gelangen sind erstellt. Die eigentlich gewollte Funktion der eigenen Spielgenerierung ist noch nicht implementiert.