

A dark blue vertical bar runs along the left edge of the page. A blue arrow-shaped graphic points to the right, containing the date. In the bottom left corner, there are several thin, curved lines in dark blue and light grey.

7.8.2016

Dokumentation

Abschlussprojekt im Fach Mobile
Computing 1,
Sommersemester 2016

Felix Wisser, Hendrik Schmidt
Fachhochschule Erfurt

Inhalt

1 Einleitung.....	2
1.1 Das Spiel Go.....	2
1.2 Intention der App.....	2
1.3 Technische Voraussetzung zur Benutzung.....	2
2 Konzept.....	2
2.1 Design und Usability.....	3
2.2 Herangehensweise.....	5
3 Umsetzung.....	5
3.1 Activities.....	6
3.2 Komponenten.....	6
4 Fazit und Ausblick.....	7
4.1 Rückblick.....	7
4.2 Offene Punkte.....	8
4.3 Ausblick.....	8

1 Einleitung

1.1 Das Spiel Go

Das Go Spiel ist ein asiatisches Brettspiel, welches mit über 4000 Jahren das älteste Brettspiel der Welt ist, welches noch nach den ursprünglichen Regeln gespielt wird. Es wird gewöhnlich mit schwarzen und weißen Steinen auf einem 19x19 Linien großen Spielfeld gespielt.¹ Das generelle Ziel des Spiels ist es mit den eigenen Steinen ein möglichst großes Gebiet freier Schnittpunkte abzugrenzen. Zusätzlich können gegnerische Steine durch vollständiges besetzen aller angrenzenden Schnittpunkte gefangen werden. In diesem Falle wird der Stein vom Spielfeld entfernt und der fangende Spieler erhält einen zusätzlichen Punkt.

Obwohl der Großteil der Go-Spieler weltweit im ostasiatischen Raum zu finden sind (insbesondere in Japan, Korea und China) gibt es in Deutschland über 30.000 Go-Spieler. Die Anfänge des Go Spiels in Deutschland gehen auf eine Go-Gruppe in Berlin zur Zeit der Weimarer Republik zurück. Inzwischen ist es möglich an nahezu jedem Wochenende des Jahres ein Go Turnier in Deutschland zu besuchen.²

1.2 Intention der App

Die Android-App, die vorläufig den Namen „GoApp“ trägt, soll gleich mehrere Funktionalitäten rund um das Spiel Go bieten. Als eine der Hauptfunktionen soll das Spielen des Spiels zwischen zwei Spielern ermöglicht werden. Das mobile Gerät dient dabei als Ersatz für das Spielbrett. Die beiden Parteien sitzen sich dabei gegenüber und setzen die virtuellen Steine auf dem abgebildeten Spielbrett in der Mitte. Funktionalitäten zum Aussetzen eines Zuges oder zum Aufgeben, sollen dabei für beide Spieler gleichermaßen gegeben sein. Weiter soll es die App ermöglichen, real gespielte Go-Spiele aufzeichnen zu können. Die Steine werden dabei parallel zum realen Spielbrett auf dem Virtuellen in der App gesetzt. Nach dem Aufzeichnen soll es dann möglich sein, die einzelnen Spielzüge nachzuvollziehen, um die eigene Spielweise und Strategie evaluieren zu können. Weiter soll es möglich sein, Go-Spiele, die über andere Medien im SGF-Format gespeichert wurden mit Hilfe der App zu laden und ggf. daran weiter zu spielen oder die einzelnen Spielzüge durchzusehen.

1.3 Technische Voraussetzung zur Benutzung

Grundsätzlich benötigt die App keine zusätzlichen Installationsschritte um ordnungsgemäß zu funktionieren. Es sei allerdings anzumerken, dass für die „Laden“ Funktionalität, .sgf Dateien im Ordner „SGF_files“ des External Storage Public Directory des Geräts liegen sollten, da andernfalls keine Spiele zum Laden zur Auswahl stehen.

2 Konzept

Um für den Fall einer Veröffentlichung der App ein größeres Publikum zu erreichen wurde entschieden, dass die App vollständig in Englischer Sprache verfasst werden sollte.

Der Übersicht halber wurde bei der Erstellung der App auf ein Hauptmenükonzept zurückgegriffen. Beim Start der App besteht die Auswahl aus den folgenden vier Menüpunkten:

- New Game
- New Record
- Load Game
- Show Tutorial

¹ 13x13 und 9x9 sind weitere Brettgrößen, auf welchen besonders Anfänger oft spielen.

² Quellen: www.dgob.de / www.go-baduk-weiqi.de

Zu erwähnen sei an dieser Stelle vor allem die klare Trennung zwischen den beiden Spielmodi „spielen“ und „aufzeichnen“ durch die jeweiligen Menüpunkte „New Game“ und „New Record“.

2.1 Design und Usability

Da sich die Intentionen der o.g. Spielmodi grundlegend unterscheiden sollte dies ebenfalls in der Struktur und dem Design der App reflektiert werden. Im ersten Fall liegt die Intention des Nutzers darin mit einem anderen Nutzer zusammen sein Endgerät als Spielfläche für ein Go Spiel zu benutzen. Im zweiten Fall hingegen möchte der Nutzer sein Endgerät zum Aufzeichnen eines parallel auf einem „realen“ Brett gespielten Spiels nutzen. Ein weiterer Anwendungsfall den aufzeichnen Spielmodus besteht im Nachspielen bereits existierender Spiele. Die im Vorfeld zur Entwicklung erstellten sog. „Mockups“ zu diesen beiden Spielmodi finden sich in den folgenden Abbildungen 1 und 2.

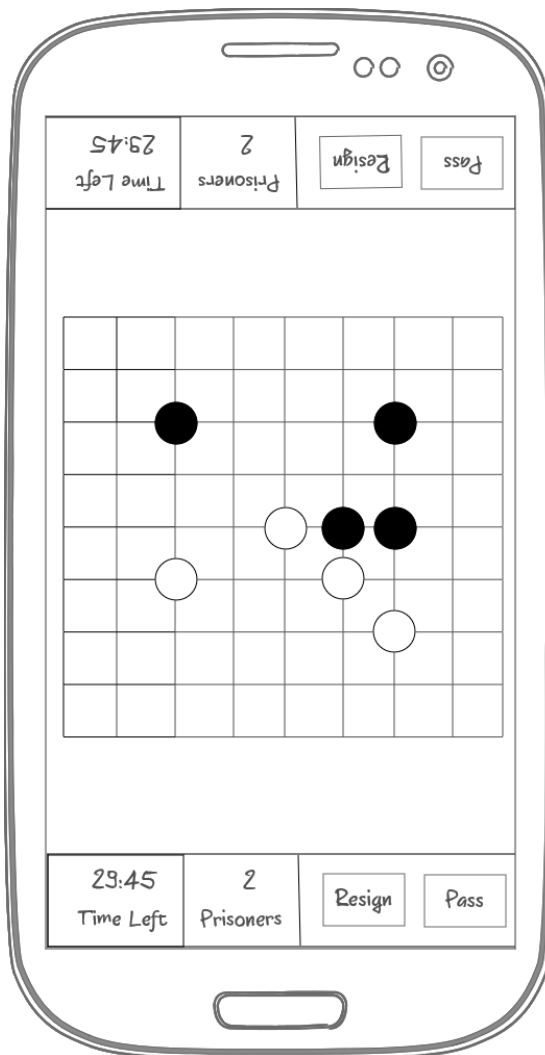


Abbildung 1

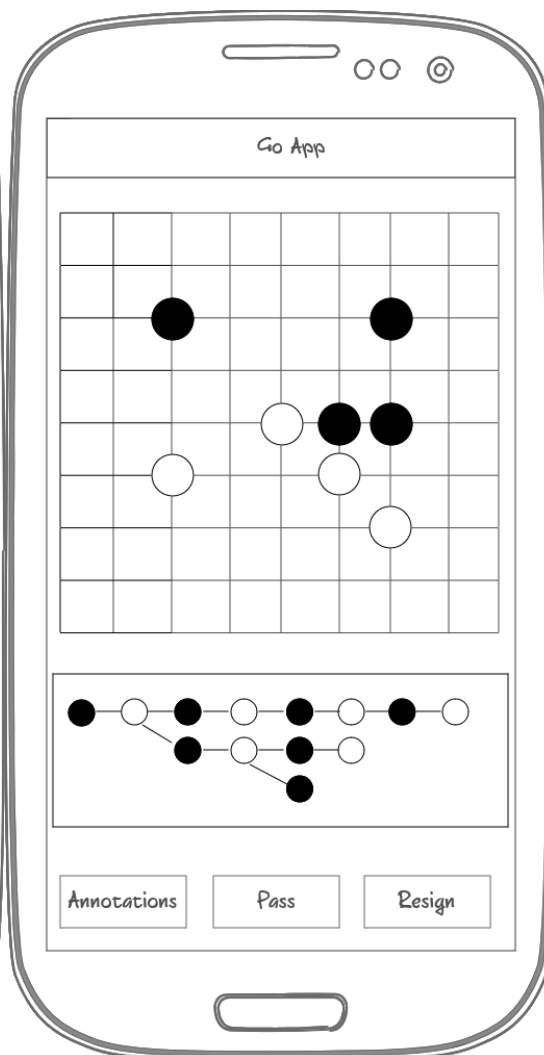


Abbildung 2

Da einige Parameter eines neu erstellten Go Spiels durchaus von Spiel zu Spiel variieren können sollte zwischen dem Hauptmenü und einem aktiven Spiel eine Konfigurationsebene eingefügt werden, welche sowohl auf den Hauptmenüpunkt „New Game“ als auch auf „New Record“ folgt. Die Mockups dieser Konfigurationsbildschirme sind in den folgenden Abbildungen 3 und 4 zu sehen. Hierbei sei zu beachten, dass im Falle des aufzeichnen Modus nur die Optionen aus Abbildung 3 zur Verfügung stehen, da andere Optionen für diesen Modus nicht notwendig sind. Im Gegensatz hierzu stehen im

spielen Modus beide Konfigurationsbildschirme zur Verfügung. Je nachdem ob der Schalter „Extended Options“ betätigt wurde werden erweiterte Konfigurationsmöglichkeiten für erfahrenere Spieler angezeigt. Diese Designentscheidung soll verhindern, dass unerfahrene Spieler von der Fülle an Auswahlmöglichkeiten abgeschreckt werden.



Abbildung 3

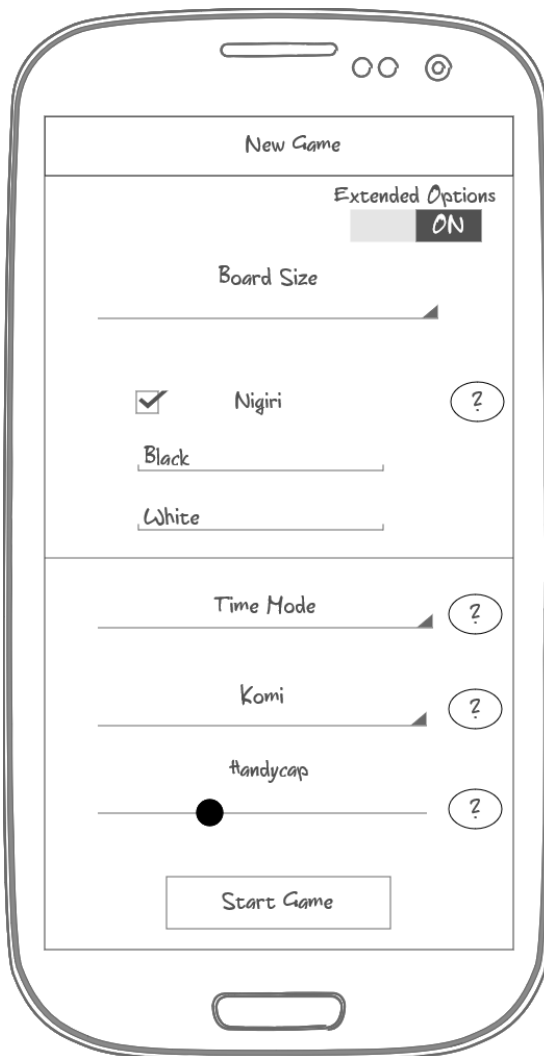


Abbildung 4

Zusätzlich zu den bereits erwähnten Spielmodi sollte es dem Nutzer möglich sein, bereits gespielte Spiele zu laden und ggfs. fortzusetzen. Diese Funktionalität soll über den o.g. Menüpunkt „Load Game“ bereitgestellt werden. Neben einer Vorschau des Endzustands des entsprechenden Spiels sollten hier ebenfalls einige Metadaten angezeigt werden, um das Spiel besser identifizieren zu können. Das Mockup zu diesem Bildschirm ist in Abbildung 5 zu finden.

Sobald ein Spiel im Modus spielen zu Ende ist sollte den Nutzern ein Ergebnisbildschirm angezeigt werden. Auf diesem sollte die Möglichkeit bestehen ein Spiel mit den selben Einstellungen zu wiederholen, in den aufzeichnen Modus zu wechseln, um das Spiel zum Beispiel zu besprechen und zum Hauptmenü zurück zu kehren. Das Mockup zu diesem Bildschirm ist in Abbildung 6 zu finden.

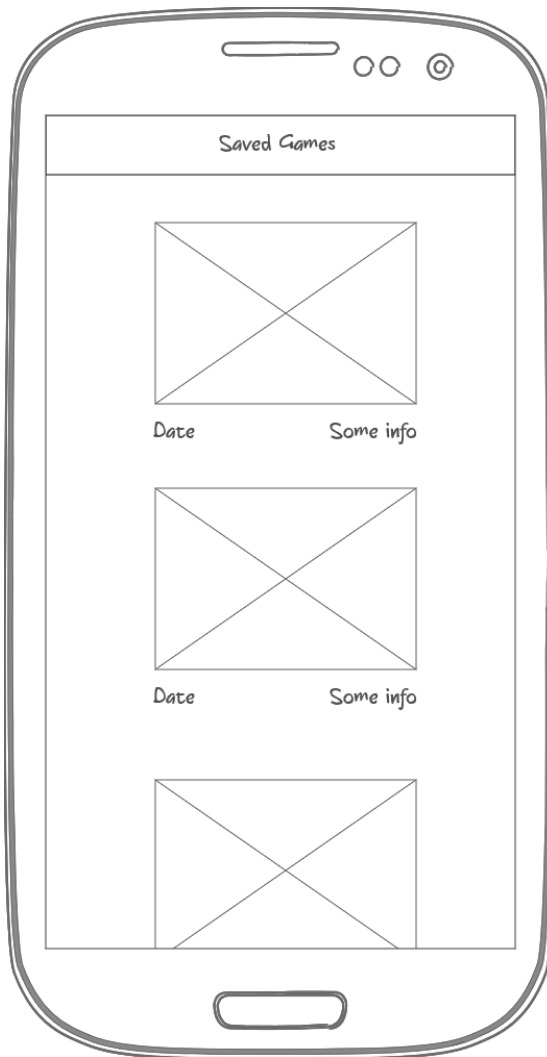


Abbildung 5

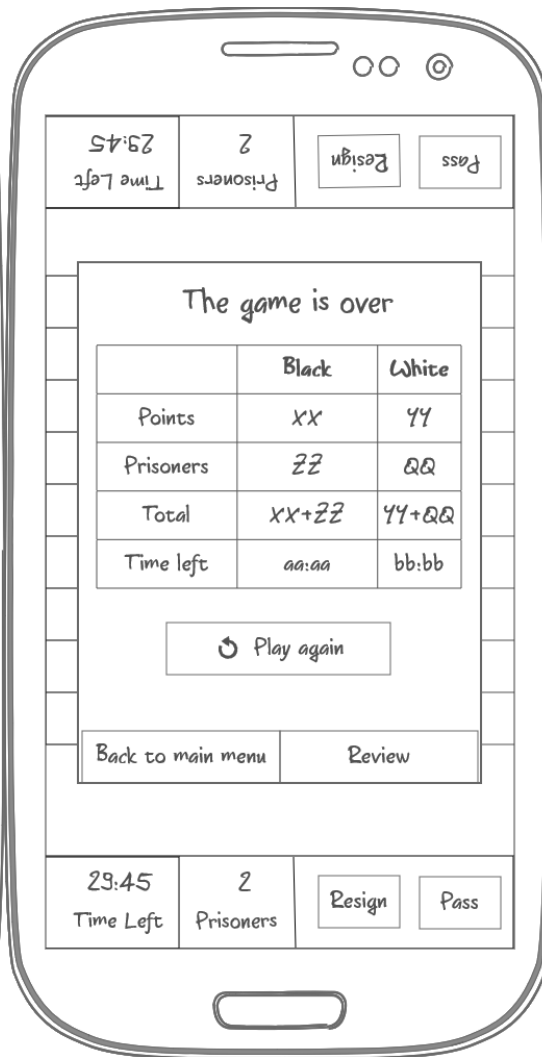


Abbildung 6

2.2 Herangehensweise

Um die Arbeit im Team möglichst produktiv zu gestalten wurde ein informeller Scrum-Prozess verwendet. Dieser bestand darin, dass in wöchentlichen Treffen der aktuelle Stand des Projekts erläutert wurde und entsprechend der erfüllten Aufgaben bzw. neuen Anforderungen neue Ziele gesteckt wurden.

Im Rahmen des Projekts wurde mit dem Versionsverwaltungssystem Git gearbeitet. Dieses wurde unter anderem gewählt um das Arbeiten an gemeinsamem Code einfacher zu gestalten und Features wie zum Beispiel Branching nutzen zu können.

Das öffentliche Git Repository des Projekts ist unter <https://github.com/kampp/GoApp.git> zu finden.

3 Umsetzung

Für die Umsetzung der App kommt das MVC-Pattern zum Einsatz. Die benötigten Funktionalitäten werden ihrer Kategorie entsprechend in Models, Views und Controllers aufgeteilt.

3.1 Activities

Für die View-Komponenten kommen im Android-System Activities zum Einsatz. Um die für die Nutzung der Funktionalitäten benötigten graphischen Oberflächen bereitzustellen, werden in der GoApp folgende Activities umgesetzt:

ActivityMain: Die Main-Activity, die auch als Einstiegspunkt in die App dient, beinhaltet lediglich Schaltflächen um den Nutzer zu den gewünschten Funktionen weiter zu leiten. Eine Oberfläche, die bei Betätigung dem Nutzer eine kurze Beschreibung sowie Anleitung, „Tutorial“ anzeigt, ist ebenfalls in dieser Activity enthalten.

ActivityNewGame: Diese Activity dient dem konfigurieren des zu spielenden Spiels. Parameter wie der Zeitmodus, Handicaps, Namen der Spieler sowie die Spielfeldgröße können hier nach Belieben konfiguriert werden. Durch eine Oberfläche werden die gesetzten Parameter bestätigt und das Spiel entsprechend den Vorgaben gestartet.

ActivityPlay: Diese Activity bildet die Hauptaufgabe der App ab. Das Spielfeld wird in der Mitte dargestellt, die spieterspezifischen Schaltflächen und Daten werden für beide Spieler am Rand angezeigt. Jeweils in gegenüberliegender Ausrichtung. In dieser Activity werden die Steine in das Spielfeld gesetzt und damit das Spiel gespielt.

ActivityRecord: Die Activity, um Spiele aufzuzeichnen lehnt sich sehr an die ActivityPlay an. Das Spielfeld befindet sich in der Mitte, jedoch sind die Schaltflächen nur auf einen Spieler, den „Aufzeichner“, ausgelegt.

BoardView: Die BoardView bildet das Spielfeld ab. Diese Subklasse der Klasse View beinhaltet Funktionen zum Zeichnen der Linien und Steine. Die View wird in den entsprechenden Activities eingebunden.

ActivityLoadGame: In dieser Activity werden alle im SGF-Format gespeicherten Spiele angezeigt. Diese können hier geladen und in der ActivityRecord weitergespielt bzw. analysiert werden.

3.2 Komponenten

Aus der Entscheidung für das MVC-Pattern heraus ergeben sich die folgenden Module, die alle ihren speziellen Teil in der App erfüllen.

Um die Daten für die einzelnen Activities bereitzustellen, werden folgende Model-Klassen implementiert.

GameMetaInformation: Die GMI-Klasse speichert alle Parameter, die für ein Spiel konfiguriert werden können. Ausgeschlossen dabei sind konkrete Fakten zum Spiel, wie z.B. die Position der Steine etc. Gespeichert werden Meta-Informationen, wie der Zeitmodus, die Spielernamen oder die Spielfeldgröße. Diese Klasse wird in der `ActivityNewGame` instanziiert und mit den gesetzten Parametern befüllt.

RunningGame: Das `RunningGame` stellt ein Spiel dar, das momentan gespielt oder analysiert wird. Hier sind die einzelnen Spielzüge in Form von `MoveNodes` gespeichert. Ein Spiel enthält jeweils eine Instanz der `GameMetaInformation`, welche die konfigurierbaren Parameter des laufenden Spiels enthält.

MoveNode: Eine Instanz der `MoveNode` enthält die Informationen zu genau einem Spielzug. Dabei enthalten sind lediglich die Position des gesetzten Steins, dessen Farbe und die Nach- und Vorgänger der Node. Ein Spiel wird als Baum aus `MoveNodes` aufgebaut, so können in einem Zug mehrere Variationen des Spiels weitergeführt werden, ohne dass ein Spielverlauf überschrieben werden muss. So können zu Evaluierungszwecken verschiedene Variationen ausprobiert werden und durch den Spieler auf deren Sinnhaftigkeit überprüft werden. Gliedert man eine `MoveNode` in den Baum ein, so fügt man dem Spiel einen weiteren Zug hinzu. Über eine Indizierung des Baumes kann auf die verschiedenen Variationen zugegriffen werden.

Um komplexe Algorithmen und Berechnungen, die in verschiedenen Activities und Punkten in der App benötigt werden, kommen Controller-Klassen zum Einsatz.

GameController: Der `GameController` stellt alle Funktionalitäten bereit, die benötigt werden, um das Spiel den Regeln entsprechend zu spielen. Hauptsächlich ist es, einen Zug zu überprüfen, ob dieser legitim ist. So ist z.B. das Setzen eines Steins, welcher nach diesem Zug gefangen wäre (auch Selbstmord oder Suicide genannt), nicht erlaubt. Die Überprüfung der Einhaltung der Regeln erfolgt nach jedem Zug und ist in einem Funktionsaufruf zusammengefasst.

TimeController: Der `TimeController` ist ausschließlich für die Berechnung der Zeit, die dem Spieler für das Spiel, oder einen Zug, zur Verfügung steht, verantwortlich. Abhängig vom ausgewählten Zeitmodus berechnet der `TimeController` die Zeit und aktualisiert die entsprechenden Anzeigen.

4 Fazit und Ausblick

4.1 Rückblick

Trotz unerwartet langer Bearbeitungszeiten in einigen Bereichen der App konnten dennoch fast alle Features in der gewünschten Form umgesetzt werden. Die Anfänglich als Optional deklarierten Features wurden zugunsten der primären Features zurückgestellt. Auch im Hinblick auf die Weiterentwicklung der GoApp sind diese Features jedoch weiter mit Eingeplant. Die Entwicklungen

an dem Projekt werden auch nach der Abgabe am 07.08.16 nicht eingestellt, was auch auf den positiven Ergebnissen des bisherigen Verlaufs basiert.

4.2 Offene Punkte

Zum Zeitpunkt der Abgabe sind nicht alle vorher geplanten Features umgesetzt. Folgende Punkte sind noch in zukünftigen Versionen der App zu gestalten:

- Anzeigen der Variationen eines Spiels innerhalb des aufzeichnen Modus
- Implementierung der Ko Regel
- Anzeigen des Endzustands gespeicherter Spiele als Bitmap
- Beenden des Spiels, sobald die Zeit eines Spielers abgelaufen ist.

4.3 Ausblick

Für die Zukunft soll es mit der App möglich sein, das Spiel mit zwei Geräten über das Internet zu spielen. Die Spieler sollen sich dabei in einer Game-Lobby finden und abwechselnd Züge durchführen. Übertragen werden dabei nur die Informationen für den aktuell durchgeführten Zug. Die Speicherstruktur der Spiele in Form eines Baumes aus MoveNodes unterstützt diese Vorgehensweise bereits. Aus den Übertragenen Informationen wird eine Instanz der Klasse MoveNode erzeugt und an den Baum angehängt.

Bei dieser Form des Spielens soll es drei Instanzen des Spiels geben. Jeweils eine auf den jeweiligen Clients und eine auf einem zentralen Server. Der Server erhält die Spielinformationen von dem Client, der den Zug durchführt und prüft den Zug auf die Einhaltung der Regeln.