Alpen-Adria-Universität Klagenfurt

Fakultät für technische Wissenschaften

Software Engineering II

LV-Nr.: 621.253, SS 2021

LV-Leiter: Dipl.-Ing. Mag. Karin Hodnigg

Thema:

Dokumentation des Gruppenprojektes „Risiko“

Projektmitglieder

Daniel Egger

Raphael Lesacher

Kristijan Rehsmann

Nikola Savovic

Peter Söllnbauer

Inhalt

Funktionsweise

Wie auch im Brettspiel „Risiko“ geht es auch in dieser Implementierung darum, die Weltherrschaft zu erlangen. Um dies zu erreichen, kann ein Spieler seine Armeen verstärken, andere Länder angreifen und seine Truppen bewegen (siehe Regelbuch). Hat ein Spieler kein einziges Land mehr in seinem Besitz, so hat er das Spiel verloren. Besitzt ein Spieler hingegen alle Länder der Welt, hat er das Spiel gewonnen.

Die Umsetzung des Spiels erfolgt als Android-Multiplayer-App. Dabei können sich mehrere Benutzer über ihr Mobiltelefon gegeneinander duellieren.

Spielablauf und Architektur

Die Applikation ist runden- und zustandsbasiert aufgebaut. Dies bedeutet, dass der Benutzer, je nachdem in welchem Zustand er sich befindet, unterschiedliche Handlungen tätigen kann.

Spielzustände:

Beim Start der App befindet sich der Benutzer in der Map-Activity. Hier kann ausgewählt werden, ob ein Spiel gestartet werden sollte, ob die Regeln angezeigt werden sollten oder ob in die Einstellungen gewechselt werden sollte.

Nach einem Klick auf den Button „Play“ verbindet sich das Endgerät des Benutzers zum Server und der Benutzer wird in die Login-Activity weitergeleitet. Hier kann der Benutzer seinen Nick Name eingeben und mit einem Klick auf den „Confirm“-Button wird der Nick Name an den Server gesendet und der Benutzer in die Lobby-Activity weitergeleitet.

In der Lobby-Activity hat der Benutzer die Möglichkeit, das Spiel zu verlassen oder das Spiel zu starten. Klickt einer der Spieler auf „Start Game“, werden alle sich in der Lobby befindlichen Spieler dem Spiel hinzugefügt und das Spiel startet.

Im ersten Zustand, dem Setup-State wählt sich jeder Spieler so lange ein Land auf der Karte aus, bis keines der Länder mehr verfügbar ist. Nach dem Setup sollte jeder Spieler die gleiche Länderanzahl in seinem Besitz haben.

Nachdem die Länder aufgeteilt wurden, beginnt das eigentliche Spiel und die zustandsbasierte Ausführung der App. Spieler 1 befindet sich nun im Draft-State. Hier wird anhand seiner Länder und seiner eingelösten Karten berechnet, wie viele Verstärkungen er zur Verfügung hat. Der Spieler kann nun seine verfügbaren Verstärkungen auf seine Länder aufteilen. Hat er die Verstärkungen aufgebraucht, wechselt das Spiel in den Attack-State.

Im Attack-State kann der Spieler ein benachbartes Land Angreifen. Dazu wird ein Würfelduell mit dem Besitzer des Nachbarlandes durchgeführt, welches über den Ausgang der Schlacht entscheidet.

Gewinnt der Angreifer, so ist das Land nun in seinem Besitz, verliert er, so verliert er Truppen. Gleiches gilt für den Verteidiger. Nach dem Angriff hat der Spieler die Möglichkeit, seine Truppen zu bewegen.

Dazu wechselt die App in den Fortify-State. In diesem State kann der Benutzer jeweils eine Armee zwischen zwei benachbarten Staaten, die in seinem Besitz sind, bewegen. Dadurch kann er sich einerseits gegen andere Angreifer besser schützen, andererseits in eine bessere Position für eigene Angriffe bringen.

Nach der Truppenverschiebung ist der nächste Spieler an der Reihe und der bisher aktive Spieler wird inaktiv, bzw. nimmt eine Zuseherrolle ein, bis er wieder an der Reihe ist. Ist der Spieler abermals am Zug, beginnt das Durchlaufen der Zustände von vorne.

Module:

Die Applikation ist in drei Module gegliedert (Server-Modul, App-Modul und Core-Modul). Dies ist der Tatsache geschuldet, dass sich das App-Modul (Android) nicht problemlos mit JaCoCo testen lässt.

Server-Modul:

Das Server-Modul ist für die Client-Server-Kommunikation essenziell. Es beinhaltet die einzelnen Nachrichtentypen, die zwischen Client und Server gesendet werden, die Logik des Servers, die Definitionen von Client und Server, sowie die Netzwerkkonfiguration.

Core-Modul:

Im Core-Modul befinden sich die einzelnen Klassen, deren Objekte im Spielablauf benötigt werden. Dazu zählen unter anderem die Klassen Country und Player.

App-Modul:

Das App-Modul besteht aus zwei wesentlichen Funktionalitäten. Zum einen befinden sich darin die Android-spezifischen Activities, welche die Schnittstelle zur GUI-bilden, andererseits befindet sich darin die Spiellogik der einzelnen Zustände sowie die Logik, welche den Spielstart sowie die Netzwerkverbindung initiieren.

Modifizierte Spielregeln - Schummeln

Die vorliegende Variante des Spiels „Risiko“ unterscheidet sich in einigen Punkten vom Brettspiel:

- Es kann jeweils nur eine Armee verschoben werden
- Armeen können nur zwischen Nachbarländern, die sich im Besitz eines Spielers befinden, verschoben werden

- Die Berechnung der Verstärkungen erfolgt nur anhand der eingelösten Karten und der in Besitz befindlichen Länder, Kontinente (Bonus) werden nicht berücksichtigt

Schummelfunktion:

Ein weiterer Unterschied zum Brettspiel ist die Schummelfunktion:

Durch starkes Schütteln des Gerätes beim Angriff/Verteidigung erhält man „gezinkte“, also höhere Würfelzahlen. Dadurch steigt die Wahrscheinlichkeit, das Würfelduell zu gewinnen.

Client-Server-Kommunikation

Die Client-Server Kommunikation erfolgt mit Kryonet. Es existiert im Projekt ein eigenes Server-Package, welches die Server-Logik hält. Mithilfe von verschiedenen Messages werden die Aktualisierungen von den Clients über den Server an die anderen Clients weitergeleitet (z.B. GUI-Updates).

Die einzelnen Messages sind:

- NameMessage: Verantwortlich für die Weiterleitung der Spielernamen.
- StartMessage: Startet ein Spieler das Spiel in der LobbyActivity, wird diese Nachricht gesendet und das Spiel startet
- Ready-Message: Verantwortlich für die Map-Activity
- TextMessage: Allgemeine Nachricht für die Weiterleitung von Text.
- TurnMessage: Teilt den Spielern mit, wann sie am Zug sind
- UpdateMessage: Wird immer gesendet, nachdem eine Veränderung eintritt (Eroberung von Ländern, GUI-Update, Verstärkung, ...)

Einen Überblick über die einzelnen Messages gibt folgender Screenshot:

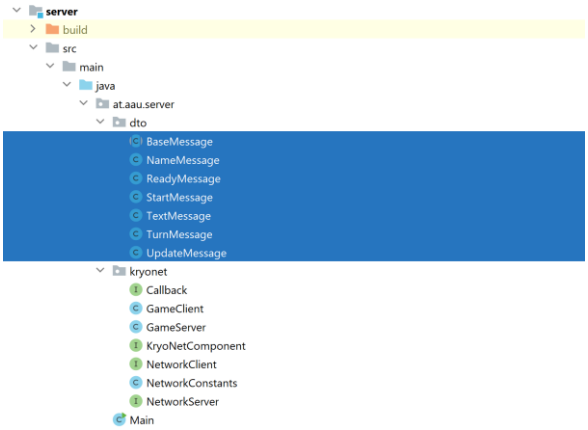


Abbildung 1: Server-Modul

In der Klasse „Main“ befindet sich zudem die Logik des Servers (Registrierung und Benützung der Messages).

Design

Die Design-Konzepte wurden vorwiegend in Figma erstellt.

https://www.figma.com/file/jah9Eyo3qYyJ5OOBpsisY7/Risiko?node-id=61%3A0

Es ist angelehnt an das Design von Hasbro, unterscheidet sich aber doch in wesentlichen Punkten.

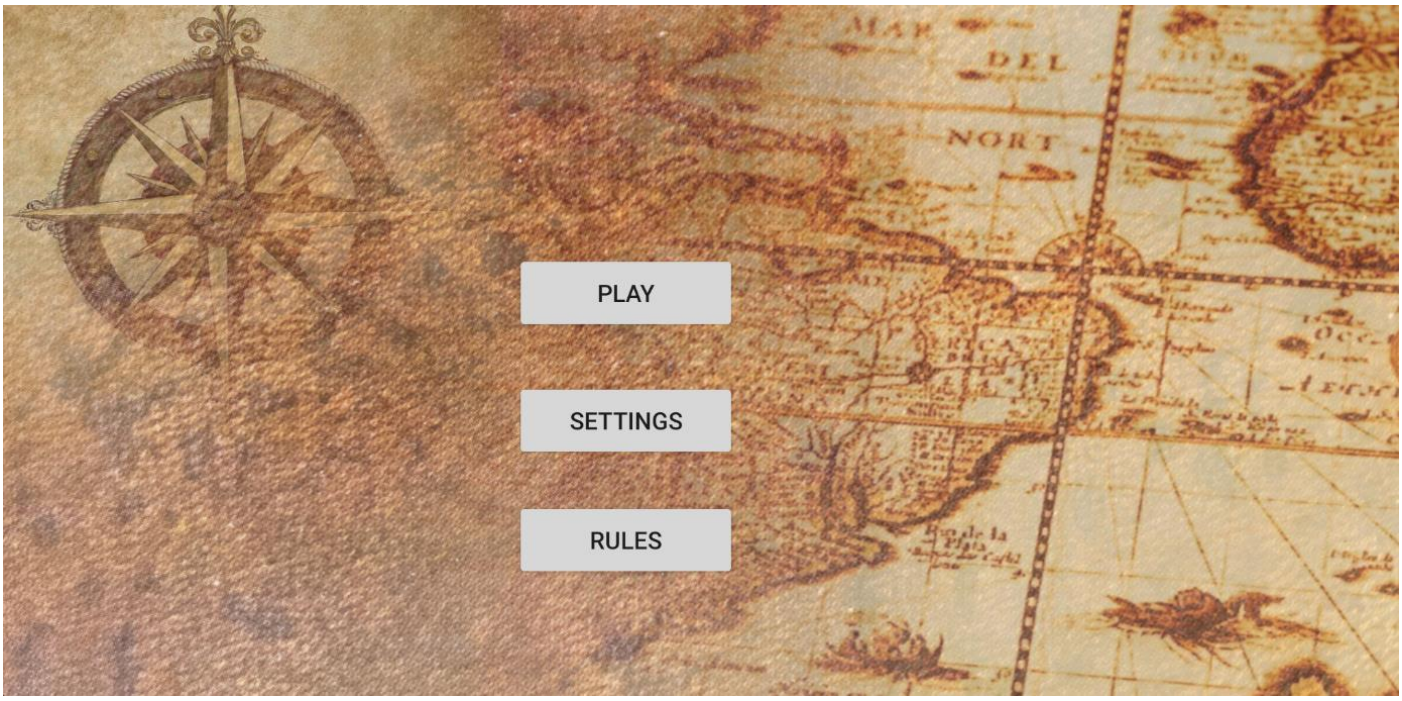


Abbildung 2: Login-Activity

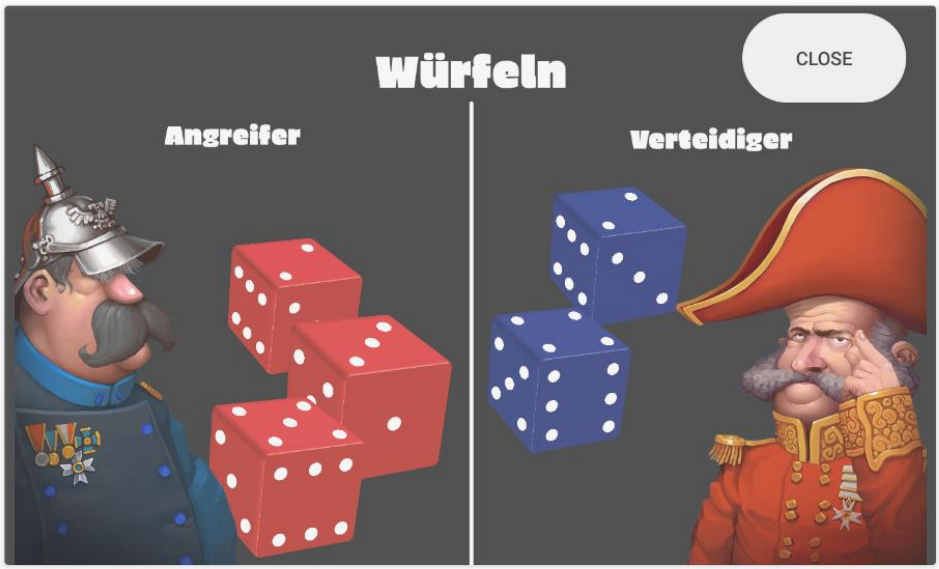


Abbildung 3: Dice-Activity

Abbildung 4: Map-Activity

Abbildung 5: Card-Activity

Verantwortlichkeiten

- Daniel Egger: Game-Setup, Server-Logik, Map-Design, Attack-State, Network-Messages - Mastermind
- Raphael Lesacher: Dice-State, Schummelfunktion, Network-Messages
- Kristijan Rehsmann: Card-Activity, Karten einlösen, Network-Messages
- Nikola Savovic: Login-Activity, Lobby
- Peter Söllnbauer: Draft-State, Fortify-State, Network-Setup, Dokumentation, NW-Messages

Weitere Informationen

Weitere Informationen zum Projekt und zur Konzeption gibt es unter:

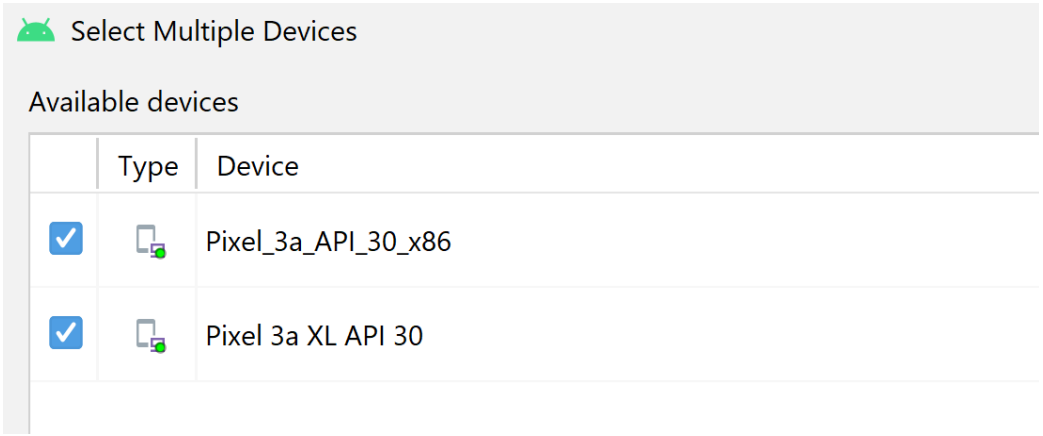
- https://github.com/castanie/SE2-Risiko
- https://www.figma.com/file/jah9Eyo3qYyJ5OOBpsisY7/Risiko?node-id=61%3A0
- https://lucid.app/lucidchart/c537f30b-7a95-404d-9d67-563916aae71d/edit?page=0_0#

Informationen zum Projektmanagement sind ersichtlich unter:

- https://github.com/castanie/SE2-Risiko/projects/1

Starten der Applikation:

- Lokal:
    - Start der Main-Klasse des Server-Moduls
    - Starten der App mit mehreren Emulatoren
        - In Android-Studio mehrere Emulatoren hinzufügen



- Über Uni-Server:
    - Start der Applikation am Handy

Informationen zu SE-Techniken

Im Laufe des Projektes wurde mit TravisCI, SonarCloud und GitHub-Actions gearbeitet. Der Umstieg auf GitHub-Actions erfolgte, da die Gratisversion von TravisCI ausgeschöpft wurde.

- https://travis-ci.com/github/castanie/SE2-Risiko
- https://sonarcloud.io/dashboard?id=castanie_SE2-Risiko
- https://github.com/castanie/SE2-Risiko/actions/workflows/android.yml

Learnings

Nach anfänglichen Schwierigkeiten, die vor allem in der Frage begründet waren, wie man so ein Projekt startet, hat sich das Team in einigen Wochen konzeptioneller Arbeit zusammengefunden und so ein Konzept für das Projekt erstellt. Nachdem die ersten Schritte erledigt waren, setzte ein positiver Trend ein, und vor allem an den verlängerten Wochenenden wurde der Arbeitsrückstand eingeholt.

Es kann schlussendlich ein ansehnliches Projekt präsentiert werden. Es ist jedoch schwierig, ohne Vorwissen so ein Projekt zu starten. In diesem Sinne sollte der Aufbau der LV evaluiert werden, und anstatt der sehr lange dauernden Einzelphase könnte man in diesem Zeitraum eine Einführung in die Konzepte der Android-Programmierung, sowie eine Einführung in die Themen „Aufbau eines Spiels“ und „Erstellung eines Projektkonzeptes“ geben.

Abbildungsverzeichnis