

# Software Engineering 1

## Abgabedokument

### Teilaufgabe 3

(ReadMe)

<b>Nachname, Vorname:</b>	Srdanovic, Nemanja
<b>Matrikelnummer:</b>	01576891
<b>E-Mail-Adresse:</b>	<a href="mailto:a01576891@unet.univie.ac.at">a01576891@unet.univie.ac.at</a>
<b>Datum:</b>	24.01.2021

## Unteraufgabe 2: ReadMe für Teilaufgabe 3

### Business Rules

- Minimale Terrain Anforderungen erfüllt – (server.rules.TerrainRuleCheck.java)
- Wasserfelder Anzahl an Kartengrenze – (server.rules.BorderRuleCheck.java)
- Keine Inseln - (server.rules.IslandRuleCheck.java)
- Burg (Terrain und Anzahl an Burgen) – (server.rules.CastleRuleCheck.java)
- Nicht mehr als zwei Player – (server.rules.PlayerRuleCheck.java)
- Nicht mehr als zwei Maps – (server.rules.TooMuchMapsSentRuleCheck.java)
- Spielzeit maximal 10 minuten (server.ServerEndpoints.checkForExpiredGames())
- Maximal 999 Spiele parallel – (server.rules.GameRuleCheck)

### Open Close Principle

Wie im Tutorial vorgeschlagen ist jede Verifikation eine eigene Klasse welche das Interface `IRule` implementiert. Durch das interface sind die Methoden welche in den Endpoints geprüft werden vorgegeben. Alle Verifikationsklassen sind in einer Liste gespeichert die jedes Game Objekt enthält, sodass spezifische daten für jedes Objekt separat gespeichert werden können. Beim aufrufen eines Endpoints wird diese Liste iteriert und die für dieses Endpoint implementierten Verifikationsmethoden aufgerufen.

### Endpoints

Alle Endpoints außer `/moves` sind implementiert.

### Bonuspunkte

Es werden keine Bonuspunkte angestrebt, da ich leider aus Zeitmangel nur ein paar Tage für die Implementierung des Servers hatte. Dadurch sind einige Methoden in der quick and dirty Weise implementiert, wessen ich mir auch bewusst bin.