

Business Rules:

- alle Regeln in Package "rules"
- Full Map Regeln in Package "rules.fullMapRules":
 - o Full Map Dimensionen Regel: "FullMapDimensionsRule.java"
 - o Anzahl der Spielerpositionen: "PlayerPositionsCountRule.java"
- Half Map Regeln in Package "rules.halfMapRules":
 - o Dimensionen: "DimensionsRule.java"
 - o Burg: "ForstExistsRule.java"
 - o Jeder Player kann ein HalfMap nur einmal an den Server schicken: "MapCanOnlyBeSentOnceRule.java"
 - o Keine Inseln: "NoIslandRule.java"
 - o W, G, M Anzahl: "TerrainTypeCountRule.java"
 - o Allgemeine Terrainanzahl: "TotalNodeCountRule.java"
 - o Wasser am Rand: "WaterEdgesRule.java"
- GameID Regel in Package "rules.gameRules"(GameExistsRule.java)
- Player Regeln in Package "rules.playerRules":
 - o Playeranzahl-Limit: "PlayerLimitRule.java"
 - o PlayerRegistration: "PlayerRegisteredRule.java"

Open-Close-Prinzip:

Zuerst habe ich nur eine IRule Interface implementiert, die von einem Beispiel aus der Vorlesung inspiriert war, aber dann, als ich mehr über das Interface Segregation Prinzip lernte, entschied ich mich, es in 2 zu teilen, je nach Anliegen - 1 für die Kartenvalidierung und 1 für die allgemeine Spiellogikvalidierung.

- "IMapRule.java" + "IGameLogicRule.java" (beide in Package "rules")
- "RuleHandler.java" in Package "rules" dient der Zentralisierung der Validierungslogik.

Endpoints:

1. Neues Spiel erstellen
2. PlayerRegistration bekommen
3. HalfMap bekommen
4. GameState schicken

Eigenschaften des Spielablaufs:

- unterstützt mehrere Spiele
- entfernt spiele, falls die Anzahl der Spiele 99 übersteigt
- validiert Business Rules
- erstellt FullMap

Ich bin aus mehreren Gründen stolz auf meine Serverimplementierung. Es ist mir gelungen, das Open-Close-Prinzip mit Hilfe des Strategie-Patterns zu gewährleisten und die Fehlerbehandlung besser zu verstehen, während ich gleichzeitig auf die anderen Best Practices wie SR, niedrige Kollision, Law of Demeter, Lesbarkeit, Verwendung von Optionals und allgemein besser strukturierten Code geachtet habe. Mein Code ist keineswegs perfekt, aber zu Beginn des Semesters hätte ich nie gedacht, dass ich so einen Code schreiben könnte.