

Business Rules

Ich habe die Business Rules in zwei Kategorien unterteilt: solche, die für die Validierung der HalfMap verantwortlich sind, und solche, die sich auf die Validierung von Player und Game beziehen.

Für die Validierung der HalfMap gibt es sechs rules:

1. **HalfMapExistsRule** überprüft, ob die HalfMap existiert.
2. **MapSizeRule** überprüft die HalfMap-Größe.
3. **FortExistsRule** überprüft, ob ein Fort in der HalfMap vorhanden ist.
4. **TerrainCountRule** überprüft die Anzahl der verschiedenen Terrain-Typen in der HalfMap.
5. **EdgeWaterCountRule** überprüft die Anzahl der Wasser Terrain an den Rändern der HalfMap.
6. **IslandCheckRule** überprüft, ob es in der HalfMap Inseln oder unpassierbare Bereiche gibt.

Für die Validierung von Player und Game gibt es vier rules:

1. **GameIdExistsRule** überprüft, ob gameId existieren.
2. **PlayerIdExistsRule** überprüft, ob playerId existieren.
3. **PlayerCountRule** überprüft die Anzahl der Player in einem Game.
4. **PlayerTurnRule** überprüft, ob ein Player nur während seines eigenen Zuges handelt.

Um das Open-Closed-Prinzip bei der Implementierung der Business Rules zu berücksichtigen, habe ich **Interfaces IBusinessRule<T> verwendet**, damit neue Regeln können leicht hinzugefügt werden, ohne den bestehenden Code zu ändern. Außerdem ist **jede Regel in einer eigenen Klasse gekapselt**, was die Unabhängigkeit und Wiederverwendbarkeit erhöht.

Funktionen des echten Servers

In meiner Implementierung wurden die ersten 4 durch das Netzwerkprotokoll vorgegebenen Endpoints unterstützt, nämlich:

1. **newGame** erstellt ein neues Game und gibt eine eindeutige gameId zurück.
2. **registerPlayer** registriert einen neuen Player für ein bestimmtes Game.
3. **receiveHalfMap** empfängt und validiert eine HalfMap von Player.
4. **sendGameState** sendet den aktuellen Game State an Player.