

Reflexion, Plan vs. Wirklichkeit

Während meiner Client Implementierung habe ich den Server ein bisschen besser verstanden, und habe mich dafür entschieden, dass ich den Server ein bisschen anders als zuvor geplant implementiere.

GameData stellt die GameState dar und beinhaltet die Daten über den Game State. Ich habe versucht die packages gleich wie im Diagram zu behalten. Ich habe 3 Controllers erstellt, GameStateController, die die GameData kontrolliert, PlayerController und FullMapHandler.

Außerdem, habe ich den anderen Klassen fast gleich wie in dem Klassendiagramm gelassen.

Das Netzwerk-Converter Konzept war sehr leicht zu implementieren wegen der schon implementierte Client.

Das Validieren der HalfMaps habe ich meistens von dem Client genommen, was ein großes Vorteil war.

Ich habe versucht die Best Practices zu folgen und auch das Single Responsibility Principle zu halten.

Die enums habe ich zusammen in einem Package gesetzt.

Die UML Klassendiagramm hat mir dazu geholfen, mir eine generelle Idee über den Klassen der Server zu machen, aber wegen unvollständige Verständnis über die Funktionsweise der Server, habe ich am Ende den Struktur geändert. Diese Änderungen waren viel schneller als beim Client.

Es war insgesamt nicht als mühsam wie die Client Implementierung und die Tests Generierung war viel einfacher.