Protokoll

Decisions

Ich habe mich entschieden die httpserverdemo als Vorlage für den HTTP-Server zu verwenden und die ado.net-repository-demo als Vorlage für die Datenbank zu nehmen

Ich habe mich entschieden sowohl die Damage calculation der Battlelogik als auch die Battlelogik selbst und die Datenbankabrufe der bestehenden Endpoints mit Unittests zu prüfen, weil sie alle sehr oft Probleme bereitet haben. Ich habe vor allen die Elementtypen und Sonderregel als Unittests getestet, da ich sie als große if-Abfrage implementiert habe, die etwas fehleranfällig war

Die Battlelogik habe ich implementiert aber noch nicht mit einem Endpoint verbunden.

Structure

Ich habe mich entschieden das Projekt in Datalogik, Models, Businesslogik und http-Server zu unterteilen. Datalogik habe ich jeweils noch in Datahandler, DataRepositories und DataModels unterteilt.

Lessons learned

Ich habe verstanden, warum man Projekt mit layered Architecture in verschiedene Projektmappen teilt, weil es wird schon ganz schön unübersichtlich je größer das Projekt wird. Ich habe gelernt, dass die Sql-Dokumentation aus der Lehrveranstaltung im ersten Semester, hilfreich ist, wenn man eine Sql-Datenbank plant.

Estimated Time

80-100 Stunden, sehr viel davon mit Recherche, wegen einiger blöden Fehler verbracht.

Gitlink

https://github.com/peziwezi/MTCG_Patrick_Rohrweckh.git

Klassendiagramm