

Reflexion, Plan vs. Wirklichkeit

Adrian Boghean - 11742914

Änderungen aufgrund von Vorschlägen, die ich während des ersten Abgabegesprächs erhalten habe:

- ClientHalfMapGenerator Klasse: validateHalfMap Methode - hat keine Parameter mehr und wird innerhalb des Konstruktors aufgerufen.
- Ich speichere nicht mehr einen Booleschen Wert, um zu sagen, ob der Schatz abgeholt wird, sondern ich frage direkt die NetworkHandler Klasse.
- PlayerData Klasse existiert nicht mehr. Alle Spielerdetails stammen aus der Main Klasse. URL und GameID auch jetzt und die gehen in die Netzwerkklassse.

Änderungen, die während der Implementierung vorgenommen wurden:

- Zunächst möchte ich erwähnen, dass ich, wie in einer Vorlesung vorgeschlagen, ein Plugin zur Berechnung von Kohäsion und Kopplung verwendet habe. Es heißt CodeMR und hat mir geholfen, einige Entscheidungen zu treffen und einige Probleme mit einigen Klassen besser zu verstehen.
- NetworkConverter wurde in mehrere kleine Klassen unterteilt, die nur ein bestimmtes Element umwandeln. So ist es auch einfacher, durch den Code zu navigieren.
- Im Kontext von TargetFinder, habe ich für mich für Strategy Pattern entschieden. FortFinding und TreasureFinding als getrennte Klassen, da ich bei der Suche nach einem neuen Ziel verschiedene Strategien verwendet habe. Am Anfang waren sie sehr unterschiedlich. Auf jeden Fall hatte ich den Vorteil, dass ich verschiedene Strategien für verschiedene Spielsituationen testen konnte, ohne die andere Phase des Spiels zu beeinflussen. Ein Nachteil war, dass ich die Zielfindungsklasse umstrukturieren musste und einige Hilfsklassen schreiben.
- ClientHalfMap Klasse: nicht mehr verwendet wird, nicht mehr existiert. Während der Implementierung habe ich gesehen, dass ich es nur brauche, um es an den Server zu senden. Ich könnte dafür die ClientFullMap Klasse genau so verwenden.
- Im Kontext von HalfMapValidation, habe ich beschlossen, die Klasse in mehrere kleinere Klassen aufzuteilen, anstatt eine große Klasse zu haben. Das war, damit ich leichter Unit-Tests schreiben, die Länge der Klasse reduzieren und sie leichter bearbeiten kann. Etwas, das ich viele Male bearbeitet habe, ist die Fortplazierungsklasse. Jetzt gibt es auch eine Unterklasse für Floodfill.
- Das am meisten veränderte Paket ist das Paket "Movement". Es gibt viele neue Klassen, viele Hilfsklassen und auch ein Interface für das Strategy Pattern.
- Im Kontext von MapAnalysis, habe ich angesichts von relevante Positionen für eine eigene Klasse, die neue Positionen vom MapAnalyser empfängt. Da der MapAnalyser bereits Daten für die View aufbereitet, könnte er gleichzeitig auch Positionen extrahieren. Sie irgendwo zu speichern ist auch sehr nützlich für die Bewegungsklassen, nicht nur für die View. Die andere Möglichkeit war, dafür Methoden in ClientFullMap zu schreiben, aber MapAnalyser überprüft bereits alle Knoten.