

Software Engineering 1

Abgabedokument

Teilaufgabe 3

(Reflexion, Plan vs. Wirklichkeit)

Nachname, Vorname:	Srdanovic, Nemanja
Matrikelnummer:	01576891
E-Mail-Adresse:	a01576891@unet.univie.ac.at
Datum:	24.01.2021

Unteraufgabe 1: Reflexion, Plan vs. Wirklichkeit

Aspekt - Open Close Principle

- **Änderung:** Jede Validierung - Methode ist zur selbstständigen Klasse geworden
- **Begründung:** Durch das Feedback und auch des vorstellen des OCP im Tutorium zur Abgabe 3 sind die Vorteile des OCP und des Nutzens von Interfaces und Abstraktion klar geworden. Dadurch habe ich die einzige Verifikation Klasse aus dem Plan in mehrere selbstständigen Klassen aufgeteilt, die durch das IRule Interface verbunden sind.

Aspekt - Kapselung vs. Kopplung

- **Änderung:** Es wurden mehrere Logikklassen zur Steuerung der Objekte gemacht, als es im ursprünglichen Plan gedacht war.
- **Begründung:** Durch das Feedback ist klar geworden, dass aus mehreren Gründen es nützlich wäre die Kontrolle der Objekte aufzuteilen bzw. aus einem GameController und einer Game Daten-Klasse mehrere unabhängige Controller zu machen. Was im Laufe der Implementierung sich als sehr hilfreich gezeigt hat.
- **Änderung:** Wo immer es möglich war, sind Methoden von public auf private geändert und nur Schnittstellen sind public geblieben
- **Begründung:** Um dem Designprinzip der Kapselung und Information Hiding gerecht zu werden, sind die meisten Methoden, welche im Plan public waren, auf private umgeändert. Die Logik der Methodenaufrufe ist dadurch in einzelnen Methoden (Schnittstellen) geblieben. Dadurch sind die Methoden nach außen nicht mehr sichtbar bzw. der interne Aufbau der Controller ist versteckt. Es ist natürlich die Möglichkeit geblieben, bei Bedarf die Methoden jederzeit sichtbar zu machen, ohne dass diese Änderung negative Auswirkungen hat, da die nur innerhalb der Klassen eingesetzt werden konnten.
- **Änderung:** Komplexe Methoden sind im Gegensatz zum Plan in kleinere unabhängige Teilprobleme zerlegt. Meistens werden dann alle diese Methoden in der Schnittstelle zur Lösung eines komplexen Problems verwendet.
- **Begründung:** Diese Methoden mussten in der Implementierung in mehrere einzelnen Methoden aufgeteilt werden, die alle wiederum zur Bewältigung einer Aufgabe eingesetzt werden. Dieser Prozess ist ganz natürlich in Laufe der Implementierung passiert, da sonst einzelne Methoden zu unübersichtlich und komplex geworden wären.