Reflexion Teilaufgabe 3:

Im Vergleich zur Version der Planungsphase hat sich der Aufbau des Server ein wenig verändert. Nachdem die vorgegebene Klasse ServerEndpoints bereitgestellt wurde, habe ich mein ServerProjekt auf dieser aufgebaut und dementsprechend erweitert. Um die Business Rules zu überprüfen wurde hier ein Interface IRule verwendet das verschiedene Klassen hervorbringt, welche auf die einzelnen Punkte der Business Rules eingehen und somit wird jede rule mit einer eigenen Klasse überprüft, so wie in dem Tutorial vorgeschlagen wurde. Eine eigene Klasse Game sowie die Klasse GameData wurde hinzugefügt, um ein einzelnes laufendes Spiel in einer eigenen Klasse abzubilden. Dazugehörig ist eben die Klasse GameData, welche nur für die Speicherung bzw. Bereitstellung der spielrelevanten Daten bereitsteht. So können in der Klasse Gamemaster die einzelnen Spiele verwaltet werden, ganz nach dem SRP.

Y-Statement 1:

- Im Kontext von der Kontroll-Klasse Gamemaster habe ich mir auf Grund von Entkopplung für eine weitere Klasse Game entschieden und eine Lösung mit einer größeren, alleinstehenden Klasse vernachlässigt, um das Prinzip von single-responsability zu erreichen und auch eine gutes Maß an Entkopplung der Klasse Gamemaster zu erreichen, unter Inkaufnahme von mehr Programmierarbeit, weil es hier eine Lösung mit wesentlich mehr Überblick bietet und die Qualität des Codes anhebt.

Y-Statement 2:

- Im Kontext von der Netzwerkklasse habe ich mich angesichts von Veränderung der Ausgangslage für die bereitgestellte ServerEndpoints Klasse entschieden und diese weiter ausgebaut und die Lösung mit einer eigenen Netzwerkklasse vernachlässigt, um auf einer existierenden Teillösung aufzubauen und nicht eine zusätzliche Klasse zu erstellen die ähnliche Arbeit geleistet hätte, unter Inkaufnahme von doppeltem Code bzw. Ausführung, weil so alles mit Netzwerkkommunikation zu Schaffende in einer Klasse komprimiert ist.

Y-Statement 3:

- Im Kontext von der Game Klasse habe ich mich angesichts von Trennung von Algorithmen und Daten für einen zusätzliche Klasse GameData entschieden und eine größere Klasse, welche Algorithmen und Daten gleichermaßen hält vernachlässigt, um diese Trennung von Aufgabenbereichen herbeizuführen, unter Inkaufnahme von Mehraufwand und einer neuen Klasse, weil somit Daten und Operationen getrennt behandelt werden können und die Klasse GameData wie eine Art Datenbank für die Klasse Game zur Verfügung steht.