Präsentation zur Qualitätssicherung

Herzlich willkommen zur Präsentation der Qualitätssicherung unseres Programms zur Simulation von wiederholten Spielen. Nach der Implementation in der letzten Phase folgte in dieser nun das Testen unseres Programmes.

Da unsere Architektur auf MVC basiert, sind wir beim Testen wie folgt vorgegangen:

* Das Model wurde anhand von JUnit Testsklassen getestet. Diese Klassen wurden zu einem beträchtlichen Teil schon während der Implementierungsphase erstellt und jetzt weiter ausgebaut und verfeinert. Insgesamt wurden so in x Testklassen y Tests erstellt. Insegesamt kommen wir im Model auf eine Testüberdeckung von xx%. Die fehlenden Prozente ergeben sich durch die Pakete Simulator, Plugin und Repository bei denen ein JUnit-Testverfahren sinnlos wären. Letztere wurden manuell geprüft.
* View und Controller wurden ebenfalls durch ausgiebige manuelle Interaktion mit der GUI getestet. Dies beinhaltet funktionale und nichtfunktionale Anforderungen wie das Erstellen/Laden/Speichern von Gruppen/Spielen/Populationen/Strategien, korrekte Ausgaben, sauberes Design.

Hier sehen sie einen Auszug aus EclEmma, der die Testüberdeckung im Model veranschaulichen soll.

Der einzige größere Bug, auf den wir gestoßen sind, ist das Speichern von Simulationsergebnissen. Hierbei war es bisher so, dass dieses die komplette SimulationHistory enthielt, welche somit auch vollständig abgespeichert wurde. Die History wird jedoch schnell sehr groß, da hier alle gespielten Duelle zwischen allen Agenten abgespeichert werde, was sich rückblickend als überflüssig erweiste.

Zum Fixen dieses Bugs waren einige Umstrukturierungen in der SimulationEngine nötig. Wir haben eine neue Klasse erstellt die nur die, für die Diagramme relevanten Daten enthält und nicht mehr die ganze Simulationhistory. Diese komprimierten Ergebnisse sind nun um den Faktor 1000 kleiner als vorher.

Sämtliche weiteren Fehler die wir gefunden haben, waren schnell behoben und sind auch nicht erwähnenswert.

Im Laufe der Qualitätssicherungen haben wir uns ein paar kleine Features ausgedacht und diese dann noch implementiert:

* Man kann Simulationen wieder aus der History entfernen in dem man auf diesen Mülleimer klickt. So kann man die History übersichtlich halten.
* Bei dem Diagramm über die Strategieverteilungen kann man nun auswählen welche Gruppen betrachtet werden sollen. Diese wählt man mit einem Haken bei der respektiven Gruppe aus.

Des Weiteren wurden einige Performance-Optimierungen vorgenommen, (Christian). Diese haben die Geschwindigkeit unserer Simulationen um den Faktor 100 erhöht.

Ein weiteres Ziel unserer Qualitätssicherung war das Finden von spieltheoretischen Erkenntnissen.

Durch die Vielfalt an Möglichkeiten die man bei der Parametrisierung einer Simulation, einer Population und eines Stufenspiels hat, ist es jedoch schwierig universelle Konzepte festzustellen. Dennoch konnten wir einige konkrete Feststellungen machen:

* Ranggleichgewicht sinnlos: Peter
* Desto höher die Vergleichswahrscheinlichkeit oder die Adaptionswahrscheinlichkeit bei einem Adaptionsmechanismus ist desto schneller stellt sich ein Gleichgewicht ein
* Bei den Erfolgsquantifizierungsalgorithmen "Payoff in the Last Adaptionstep" und "Sliding Mean" stellt sich viel schneller ein Gleichgewicht ein als bei "Total Capital" und "Total Payoff"