Präsentation zur Qualitätssicherung

Herzlich willkommen zur Präsentation der Qualitätssicherung unseres Programms zur Simulation von wiederholten Spielen. Nachdem wir letztes Mal das Programm fertig implementiert hatten, mussten wir es jetzt testen.

Weil wir in unserem Programm die MVC Architektur benutzen, sind wir wie folgt vorgegangen:

* Das Model wurde anhand von JUnit Testsklassen getestet. Diese Klassen wurde zu einem beträchtlichen Teil schon während der Implementierungsphase erstellt und jetzt weiter ausgebaut und verfeinert. Insgesamt wurden so in x Testklassen y Tests erstellt. (Manche Testklassen überdecken auch mehrere Klassen aus dem Model, so dass es insgesamt weniger Testklassen als entsprechende Klassen im Model gibt.) Die Testabdeckung im Model liegt bei +-95% (kann noch ändern). Die fehlenden Prozente ergeben sich durch ...
* View und Controller wurden durch ausgiebiges manuelles Testen der GUI getestet, das beinhaltet Funktionalitäten wie das Erstellen/Laden/Speichern von Gruppen/Spielen/Populationen/Strategien, korrekte Ausgaben, sauberes Design.

Der einzige größere Bug auf den wir gestoßen sind umfasst das Speichern von Simulationsergebnissen. Hierbei war es bisher so, dass diese die komplette SimulationHistory enthielt, die somit auch vollständig abgespeichert wurde. Die History wird jedoch schnell sehr groß, da hier alle gespielten Duelle zwischen jeweils zwei Agenten abgespeichert werden. Bei größeren Simulationen ergaben sich somit Dateien die durchaus im Gigabyte-Bereich liegen konnten, was natürlich sehr unpraktisch ist.

Zum Fixen dieses Bugs waren einige Umstrukturierungen in der SimulationEngine nötig. Es musste eine neue Klasse erstellt werden, die nur die für die Ausgabe der Diagramme relevanten Informationen enthält. ... Diese wird dann abgespeichert.

Kommen wir nun zu den Erkenntnissen die wir aus dem Programm gewonnen haben.

Durch die sehr vielen Möglichkeiten die man bei der Parametrisierung einer Simulation hat, ist es schwierig universelle Konzepte festzustellen. Die gesamte Simulation ist natürlich in erster Linie abhängig davon, welches Stufenspiel gespielt wird. Wählt man die Auszahlungen zum Beispiel so, dass ein Agent bei Kooperation eine viel höhere Auszahlung bekommt, so sind Agenten mit kooperativen Strategien logischerweise klar im Vorteil. Auch die gewählte Population ist natürlich ausschlaggebend.

....