2.3 Datenbank und Binär-Codierung

Da die Anzahl der möglichen Spielzustände sehr hoch ist wird eine effiziente Speicherstruktur für den Q-Player sehr wichtig. Es müssen beliebig viele Zustände gespeichert werden können und der Zugriff auf die Zustände muss sehr schnell ablaufen, da der Q-Player während eines Spieldurchlaufs in jedem Zug sowohl für diesen Zug, als auch für die nächsten Züge Abfragen an die Datenbank generiert. Aus diesem Grund wurde eine HashMap gewählt, die intern von der Klasse *Q\_DB* verwendet wird. Der Schlüssel dieser HashMap ist ein Spielzustand. Der Spielzustand wird von der Klasse *Game* als 2-dimensionales Int Array gespeichert. Da dieser Datentyp sich aufgrund eine fehlenden Hash- und Equalsfunktion nicht als Schlüssel eignet, wird jeder Spielzustand in einen *Array2DWrapper* umgewandelt, der diese Funktionen implementiert. Als Wert für die HashMap wird wiederrum eine neue Datenstruktur verwendet, die den die Aktion und den Wert für diese Aktion enthält. Die Aktion ist die Spalte in die geworfen wird und der Wert ist abhängig von der internen Bewertung des Q-Players und wird von diesem verändert. Die dafür verwendete Datenstruktur ist ebenfalls eine HashMap. Es könnte auch eine ArrayList oder ähnliches verwendet werden, da hier nur eine maximale Anzahl von sieben Aktionen und damit sieben Einträgen möglich ist.

Die Datenbank implementiert weiterhin folgenden Methoden mit denen der Q-Player seine Werte abfragen und verändern kann:

* *get(..)*, *containsState(..)* für Abfragen an die Datenbank
* *getValueOfStateAndAction(..)*: Liefert für einen bestimmten Zustand und die Aktion, die darauf ausgeführt werden soll, den Wert dieser Aktion zurück.
* *put(..)* und *update(..)* zum Verändern der Datenbank.
* Sowie Funktionen zum Speichern und Laden der Datenbank über die Serialisierung der HashMap.

Das Speichern und Laden der HashMap wurde inbesondere dafür verwendet, die Datenbank für den Q-Player nach einer gewissen Anzahl an Trainingsiterationen zu speichern. Damit kann das Trainings zum späteren Zeitpunkt fortgesetzt werden, indem die Datenbank wieder geladen wird.