Umsetzung MVVM

Für das zu erstellende Programm wird folgende Architektur auf Basis von MVVM gewählt:

**EventHandling**

**Utilities**

**ViewModel**

**Model**

**Game**

**Set**

**Move**

**GameField**

**Communication Server**

**KI**

**View**

**MainUIController**

**FXML**

Properties

**GameController**

**Log**

**DBConnection**

**View**

Der View nach MVVM wird durch die FXML „MainUI.fxml“ und den „MainUIController“ abgebildet. In der FXML ist rein deklarativ der Aufbau des UIs beschrieben, während der UIController die Logik des UIs beinhaltet. Beide Elemente sind von daher eng mit einander verbunden.

**ViewModel**

Das ViewModel nach MVVM wird durch die Klasse GameController umgesetzt. Der GameController stellt Methoden und Properties für den View zur Verfügung. Der View verbindet sich mittels Data Binding mit den Properties wodurch Änderungen der Properties im GameController und View synchronisiert werden. Der GameController kontrolliert das Model und steuert den Programmablauf.

**Model**

Das Model wird durch die Klassen Game, Set, Move und GameField gebildet. Das ViewModel greift primär über Game auf das gesamte Model zu. Ein Game beinhaltet mehrere Sets. Ein Set beinhaltet mehrere Moves und ein GameField.

Utilities

Zur Unterstützung des ViewModel werden unter anderem folgende Klassen implementier:

* DBConnection – zum Verwalten der Datenbankverbindung
* Log – zur Erstellung eines zentralen Logs
* CommunicationServer – zum Verwalten der Serververbindung
* KI – zur Berechnung von Zügen und Gewinnerkennung
* EventHandling – zur Übermittlung von asynchronen Nachrichten an das ViewModel