





Erläuterungen zu der Architektur

Die Software ist wie folgt aufgebaut.

Momentan sind drei Packages vorhanden, "figures", "game" und "schach", diese sollten je nach Funktion in eine der drei Packages aufgeteilt werden. Für die Gui Elemente ist ein 4. Package geplant.

In figures liegen alle Figuren mit Klassen vor und in den jeweiligen Klassen ist definiert, was ein gültiger Zug ist und was nicht.

Im package "game" liegen alle relevanten Klassen für das Spiel vor. Dazu gehört, das Board, die Oberklasse der Figuren "Figures", eine Spielerklasse, die für spätere Iterationen gebraucht wird, eine Zugklasse mit der Züge gespeichert werden können für das spätere Anzeigen der Notation und für den en passant Zug und die StartGame Klasse, die die Eingaben konvertiert und weitergibt, falls diese gültig sind und auch auf durchgängig auf Matt, Schach etc prüft und die jeweiligen Ausgaben ausgibt.

Die Board Klasse beinhaltet ein Array auf dem die Figuren sich bewegen und speichert geschlagene Figuren und die Figures Klasse dient als Mediator zwischen den einzelnen Figurtypen und dem Schachbrett.

Im "schach" package liegen dann Main, GuiMain und Console vor, also alle übergeordneten Klassen.

Bei einem typischen Schachzug wird die Eingabe in StartGame umgewandelt und der Figur, die im Feldarray des Boards gespeichert ist, ein move-command übergeben. In der Figur wird geprüft ob der Zug gültig ist und dann eine Anweisung an das Board geschickt, mit der das Spielfeld verändert wird. Zum Schluss wird geprüft ob eine Schach-, Matt- oder Patt-Situation vorliegt und der gültige Zug in die Liste der Züge eingefügt.

Für weitere Funktionen ist geplant, diese ins game package zu legen und diese dann in StartGame aufzurufen, welches unsere Schnittstelle darstellt.