

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

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

Den Figuren haben wir ein eigenes Package gegeben, da ihr Funktionen nichts mit den restlichen Klassen zu tun haben und nur von denen aufgerufen werden.

Sie liegen im Allgemeinen auf einer niedrigeren Ebene als andere Klassen.

Im package "game" liegen alle relevanten Klassen für das Spiel vor. Dazu gehört, das Board, die Oberklasse der Figuren "Figures", eine Spielerklasse, eine Timeklasse, die AI-Klasse und deren AiCalcs-Klasse, GuiCalcs mit Berechnungsmethoden der Gui-Klassen, eine Zugklasse mit der Züge gespeichert werden können für das 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 hier angesiedelten Klassen bilden die mittlere Ebene unseres Programms, da es Klassen sind, die häufig aufgerufen werden, aber in der Hierarchie noch über den Klassen des "figures" Packages stehen.

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.

Die AI Klasse ist das Grundgerüst der AI und die AICalcs beinhaltet Methoden, die von dieser zur Positionsbewertung verwendet werden.

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

Diese Klassen liegen in einem eigenen Package, damit alle Klassen aus allen Packages in gleicher Weise auf sie zugreifen können und Sie im Allgemeinen auf einer höheren Ebene liegen, als der Rest der Klassen.

Alle gezeichneten Elemente der Gui liegen in der Klasse GuiView vor, GuiCalcs ist eine Klasse um Berechnungen, die in der Gui vorkommen auszulagern.

GuiController steuert die Funktionen der einzelnen Elemente , die in GuiView gezeichnet werden.

Die AI liegt , wie alle spiel-relevanten Klassen, im package game vor und kann in der Gui oder in der Konsole jeweils ausgewählt werden.

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.

Wenn PlayervsAI ausgewählt wurde, wird das board mit diesem vorgenommenen Zug an ein erstelltes KI-Objekt übergeben.

Bei Tiefe < 1 führt dieses KI-Objekt einen random move aus, bei einer höheren Tiefe führt diese eine minmax-Suche mit alpha-beta-pruning durch, in der alle möglichen Züge jeweils für die AI und den Gegner simuliert werden, und wählt den Zug, der für sie eine höhere Chance besitzt das Spiel für sich zu entscheiden.

Weitere Features können als Klassen im game package erstellt werden und dann in StartGame für die Konsole und in GuiController und im GuiView für die GUI aufgerufen werden.





