

Architektur des Schachprogramms

Der Code ist grundsätzlich nach der Model-View-Controller Struktur aufgebaut. Der Aufbau des Schachfeldes befindet sich in dem Package Model, das View Package gibt das angepasste Schachfeld auf der Konsole oder der GUI aus und das Controller Package aktualisieren das Model (vgl. Abbildung 1). Die Klasse ViewContainer des View Packages dient hierbei als Interface für die View der Konsole als auch für die View der GUI-Version.

Die Main Klassen befinden sich in dem Schach Package. Über die Klasse ConsoleMain werden die einzelnen Komponenten (Model, ConsoleView und ConsoleController) des konsolenbasierten Spiels erstellt und miteinander verbunden. Das GUI-basierte Spiel wird über die Klasse GuiMain gelauncht. Hierbei gibt es jeweils eine View- und Controller-Klasse für das Auswahlfenster (GuiView und GuiController) sowie für das eigentliche Schachfenster (GuiViewBoard und GuiControllerBoard).

Im Weiteren werden die einzelnen MVC-Pattern Komponenten aufgeführt und kurz beschrieben.

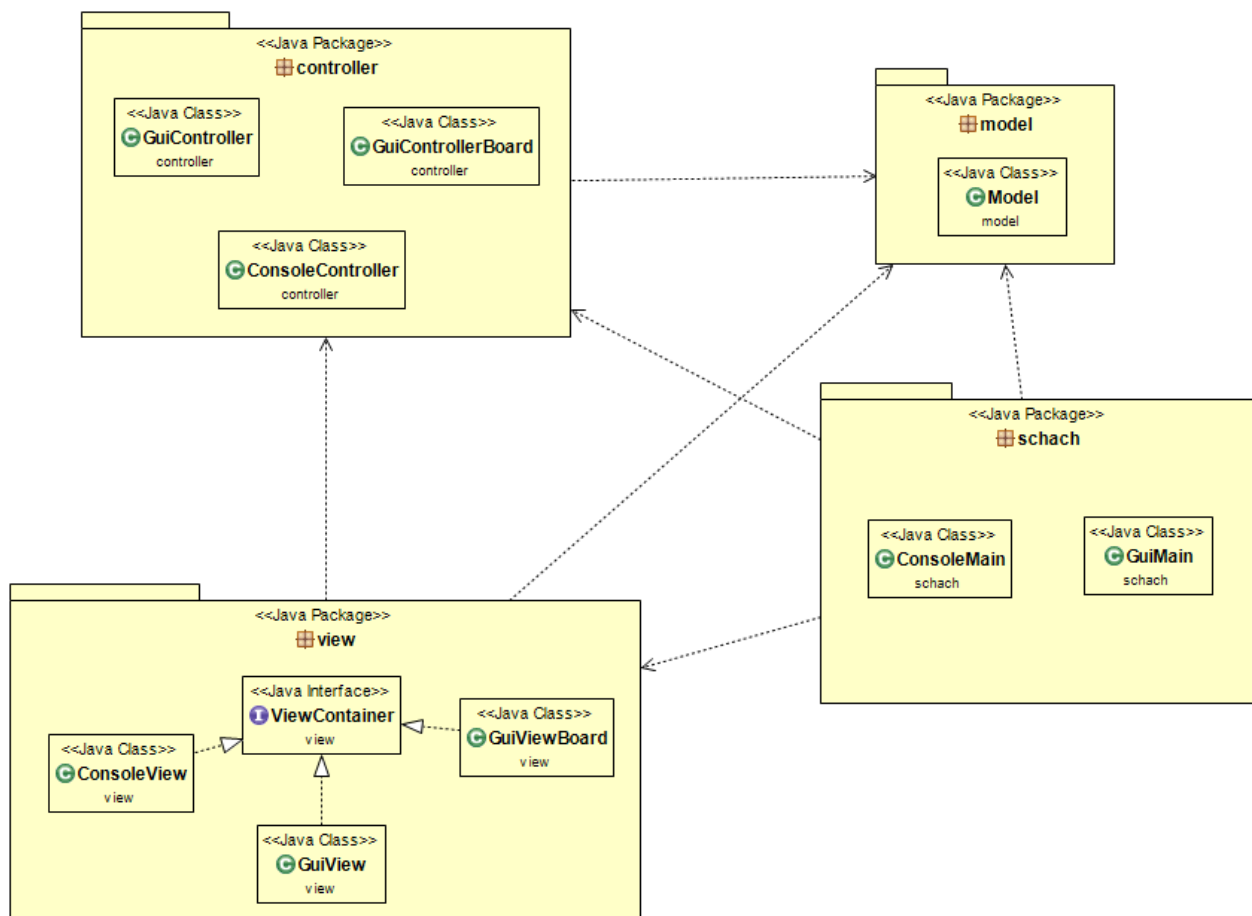


Abbildung 1: MVC-Pattern des Programms

Das Package Model beinhaltet die gleichnamige und essentielle Klasse Model, in der das Schachfeld als zweidimensionales Array gespeichert ist und bei jedem Spielzug aktualisiert wird (vgl. Abbildung 2). Es wurde ein Array aus chars gewählt, da dadurch weniger Speicher verbraucht wird und dies zu einer besseren Performanz führen kann. Darüber hinaus ist durch das frontendübergreifende Model ein Zugriff sowohl von den GUI-basierten View und Controller Klassen, als auch von den konsolenbasierten Klassen gegeben.

Ebenso befinden sich dort die Bilder der Schachfiguren, auf die in der GUI zugegriffen wird, sowie die Liste der geschlagenen Figuren.

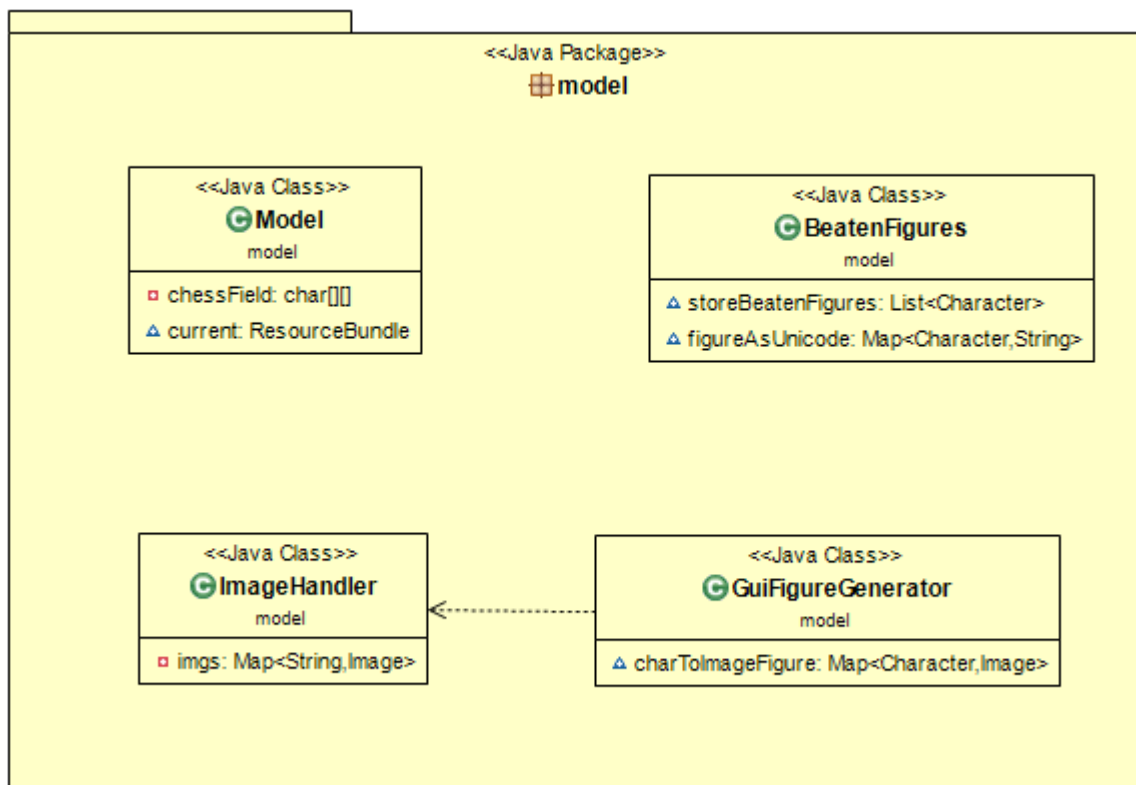


Abbildung 2: Package Model

Das View Package beinhaltet das gleichnamige Interface, auf welches die View Klasse der Konsole (ConsoleView) sowie die View Klassen der GUI (GuiView und GuiViewBoard) zugreifen (vgl. Abbildung 3). Dabei dient GuiView der Darstellung des Startfensters und GuiViewBoard der Darstellung des Spielfensters. Neben den drei essentiellen Klassen befinden sich dort ebenso Hilfsklassen wie BoardComponents und ButtonsActivated, die die visuellen Komponenten des Spielfensters beinhalten, sowie Klassen zur Erstellung von kurzweiligen Auswahlfenstern und Meldungen.

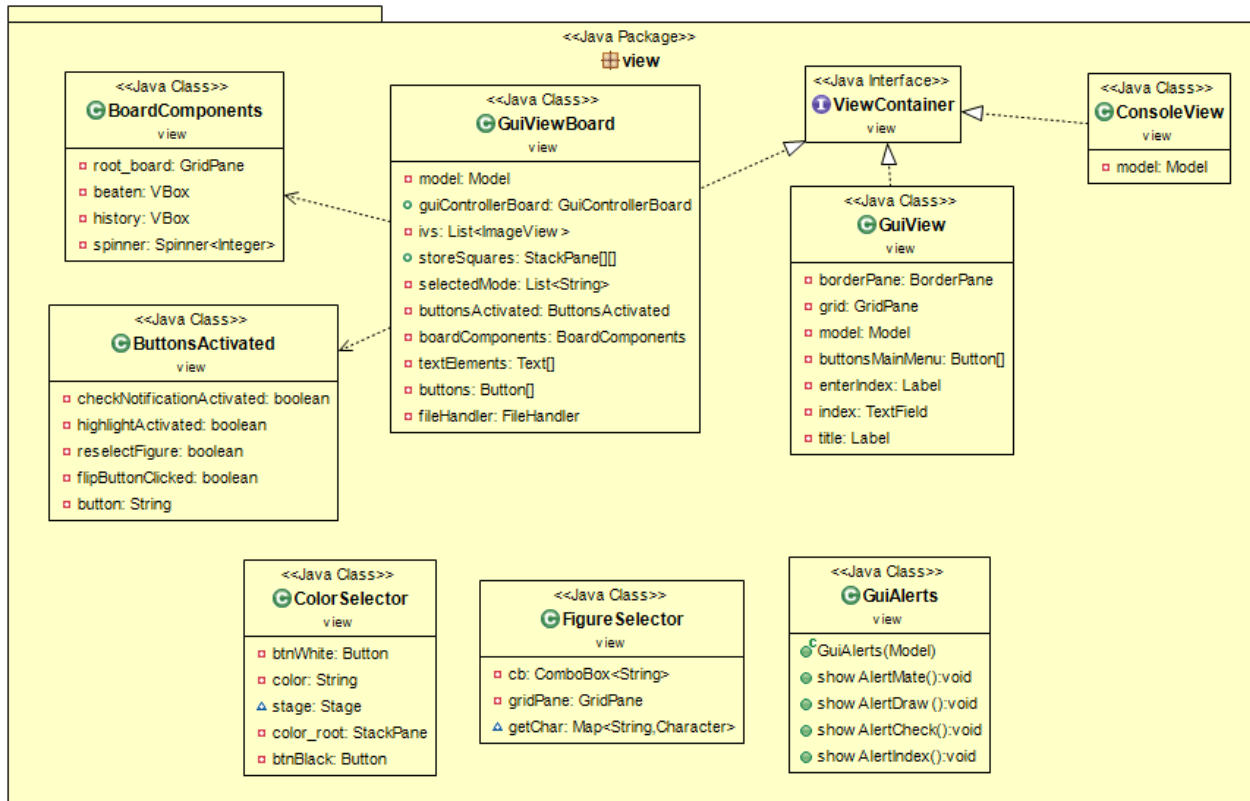


Abbildung 3: Package View

Das Package Controller (vgl. Abbildung 4) beinhaltet analog zur View ebenfalls 3 Controller Klassen für die konsolenbasierte Version (ConsoleController), das Startfenster der GUI (GuiController) und das Spielfenster der GUI (GuiControllerBoard).

GuiControllerBoard und ConsoleController können dann nach jedem Spielzug unter Zuhilfenahme von Hilfsklassen für die syntaxkonforme Überprüfung (Move), regelkonforme Überprüfung (Figure inkl. der einzelnen FigurenKlassen) und der Prüfung der Ausgangssituationen (Draw, Check, Movegenerator) sowie der KI für das Spiel gegen den Computer (AI, AIThread), das frontendübergreifende Model aktualisieren. Diese Hilfsklassen sind ebenfalls frontendübergreifend, somit war eine einfachere Implementierung der GUI möglich. Außerdem ist dadurch auch im Falle von einer fehlerhaften Codeausführung eine schnelle und simplere Problemfindung innerhalb der Hilfsklassen gegeben.

Mithilfe dieses Patterns kann ebenso eine unproblematische Erweiterbarkeit von weiteren Zusatzfeatures stattfinden. Zur Umsetzung kann lediglich eine neue Klasse im Controller Package erstellt werden, welche dann von den Controllern der beiden Frontends gleichermaßen genutzt werden kann.

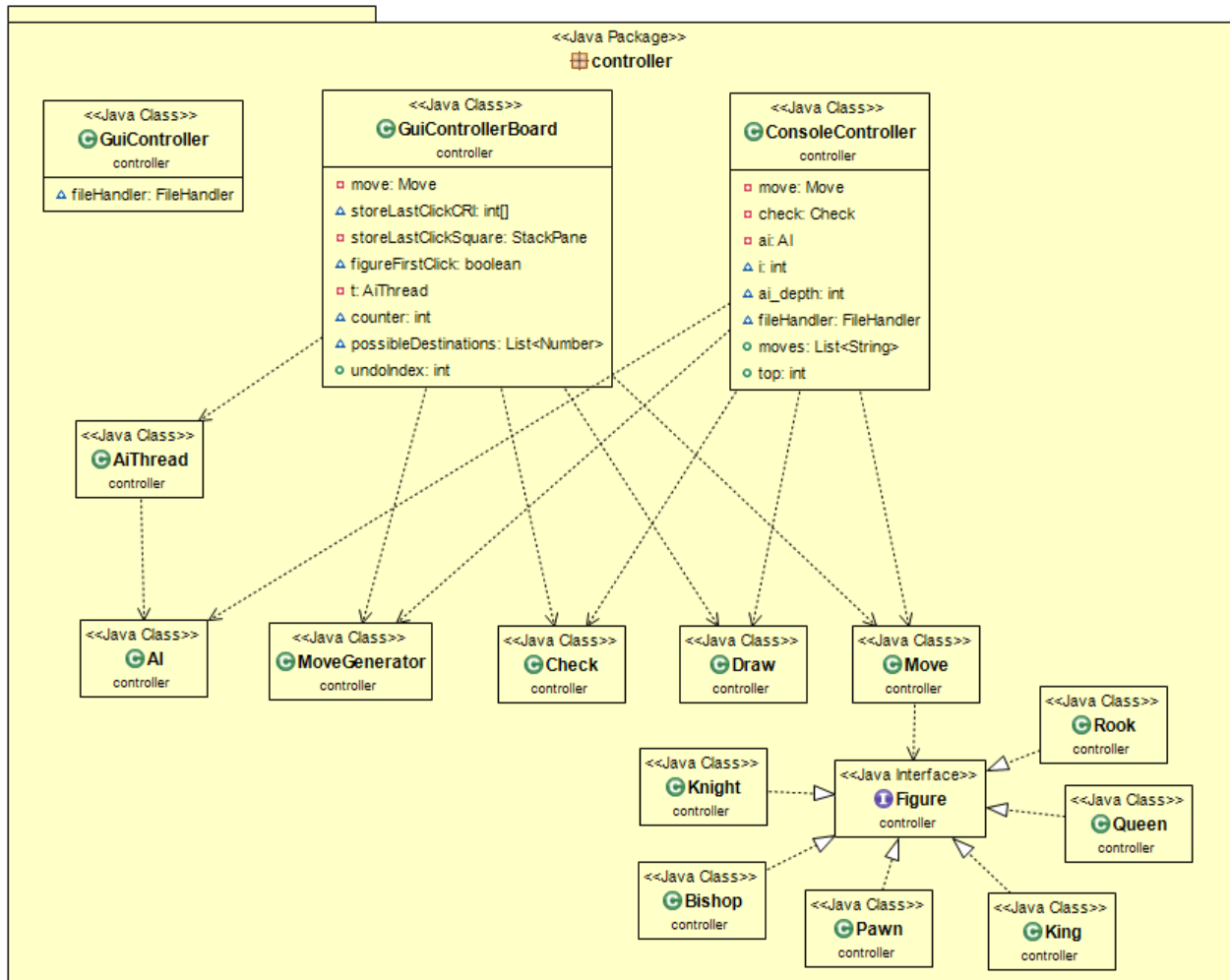


Abbildung 4: Package Controller