Architektur

Erklärung / Begründung der Architektur

Als grobes Designpattern haben wir das Model-View-Controller (MVC) Design Pattern gewählt und grundsätzlich versuchen wir möglichst einen objektorientierten Ansatz umzusetzen.

Controller

Hier befinden sich Klassen, die den Ablauf des Spiels durch Inputs und Regeln bestimmen

- Class Input: liest den Input des Users und steuert den Ablauf des Spiels durch eine Ein-, und Ausgaberoutine
- Class Check: beinhaltet die Regeln zu Schach- und Schachmatt-Situationen und regelt dadurch den Spielverlauf

Package AI:

- Class BoardValueNode: dient dafür, eine Art Baumstruktur darzustellen bei dem jedes Piece-Objekt als einzelnes Blatt gilt. Es werden hier dann die verschiedenen Werte für die jeweiligen Pieces vergeben und verglichen. Pawn hat zum Beispiel nur einen Wert von +-10, die Queen dagegen jedoch schon einen Wert von +-90. Außerdem befinden sich die verschiedenen Methoden zum MinMax-Algorithmus und zum Alpha-Beta-Pruning-Algorithmus hier.
- Class MinMaxAi: Hier wird der MinMax-Algorithmus aus BoardValueNode aufgerufen wodurch ein neuer Move in Form eines Strings generiert wird.
- Class SimpleAi: Dies ist eine AI die quasi nur einen zufälligen möglichen Move heraussucht und diesen dann ausführt. Die Klasse BoardValueNode wird hierbei in keiner weise berücksichtigt.

Model

Hier finden sich die Klassen, die nach einem objektorientierten Vorgehen die Objekte des Schachs und ihre Funktionen repräsentieren

- Class Board: Repräsentiert das Schachbrett an sich und beinhaltet Methoden, die den Spielverlauf auf dem Brett darstellen und steuern.
- Class Square: Stellt die einzelnen Felder auf dem Schachbrett dar, diese können verschiedene Werte aufweisen, wie zum Beispiel ob eine Spielfigur auf dem Feld steht.
- Interface Piece: Bildet ein Interface, welches die einzelnen unterschiedlichen Figuren zusammenfasst und gemeinsame Funktionen und Verhaltensweisen bündelt.
- Class King, Queen, Bishop, Knight, Rook, Pawn: Sind die konkreten einzelnen unterschiedlichen Figuren.

View

Hier liegen die verschiedenen Klassen für die grafische Benutzeroberfläche. Es gibt verschiedene Klassen um die jeweiligen Elemente auf dem GameScreen (FXML) richtig steuern zu können.

- Class GuiMain: Sorgt dafür, dass die JavaFX GUI gestartet wird und die Übergänge zwischen den verschiedenen Szenen durchgeführt werden.
- Class StartMenu: Wenn das Spiel gestartet wird, öffnet sich zunächst ein Start-Menü, in dem der Spieler alle Eigenschaften des Spiels einstellen kann. Diese Klasse regelt dabei die Aktionen, die ausgeführt werden, falls der Spieler etwas an den Einstellungen verändert.
- Class GameScreen: Hier werden die Funktionen und verschiedenen Komponenten des Hauptspielfensters aufgerufen und verwalten. Alle weiteren Funktionen, wie zum Beispiel die Schachuhr oder die MoveHistory werden hier quasi das erste Mal aufgerufen.
- Interface ChessBoardController: Ist das Interface, in dem die verschiedenen Gemeinsamkeiten der ChessBoardComputer und der ChessBoardHuman beschrieben werden. Beispiele hierzu sind das Einfärben der nächsten möglichen Felder, auf die ein Piece gesetzt werden kann, oder das Rotieren des Spielfeldes.
- Class ChessBoardHuman: In dieser Klasse werden die verschiedenen Inputs der Spieler gehandhabt und auf dem Spielbrett ausgeführt. Spielt man gegen den Computer wird die Klasse ChessBoardComputer benötigt.
- Class ChessBoardComputer: Diese Klasse dient dazu die richtige KI auszuwählen und deren Züge dann auch durchzuführen. Die Berechnung für die Züge der KI findet in dem oben beschriebenen Package AI unter Controller statt. Die ChessBoardComputer ist außerdem nicht nur dafür zuständig die Pieces des KI zu bewegen, sondern auch zum Beispiel für deren Promotion.
- Class LastMoveController: Hier werden die verschiedenen Eigenschaften der Anzeige für die zuletzt getätigten Züge erstellt. Diese werden an der rechten Seite des Spielfeldes ausgegeben.
- Class ClockController: Diese Klasse dient dazu die Funktionalitäten der Schachuhr am linken Bildschirmrand zu kontrollieren. Die Schachuhr funktioniert bei uns komplett unabhängig vom Spielfeld. Näheres dazu in der Bedienungsanleitung.
- Class CemeteryController: Hier wird dafür gesorgt, dass die Pieces die bereits auf dem Spielfeld geschlagen worden sind, auf dem Cemetery angezeigt werden. Dieser befindet sich unter dem Schachtbrett in der GUI.

Resources

Hier liegen die FXML auf die mit JavaFX zugegriffen werden kann und auf dem Bildschirm als eigenes Fenster geöffnet werden kann.

- gameScreen: Baumstruktur des GameScreens auf dem sich die Menü Bar, das Spielfeld, die Schachuhr, der Friedhof und die LastMove-Komponente befindet.
- startMenu: Baumstruktur des Start Menüs, auf dem alle Einstellungen getätigt werden können, bevor ein Spiel gestartet wird.
- assets: Hier befinden sich die Bilder für die verschiedenen Pieces
- images: Hier befinden sich die Vektorgrafiken der Pieces, die auch auf dem Spielfeld verwendet werden.
- languages: Hier befinden sich die Sprachoptionen für die beiden Sprachen Englisch und Deutsch. Diese können jederzeit über die Menü Bar eingestellt werden.