DHBW
Duale Hochschule
Baden-Württemberg
Karlsruhe

Kleine Spielesammlung von 4-Gewinnt Versionen

Bachelor of Science

Studiengang Informationstechnik

an der

Dualen Hochschule Baden-Württemberg Karlsruhe

von

Andreas Schmider

Abgabedatum 30. April 2022

Bearbeitungszeitraum	2 Semester
Kurs	TINF19B3

Erklärung

Wir versichern hiermit, dass wir unsere Studienarbeit mit dem Thema:

Kleine Spielesammlung von 4-Gewinnt Versionen

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt haben. Wir versichern zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Karlsruhe, 30. April 2022

Ort, Datum Unterschrift

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Beschreibung der Software

Beschreibung der Funktionalität
Aus einem alten Projekt existiert schon ein Connect6 Spiel, das in der Konsole gespielt wird. Dabei gibt es einige Einstellungen die während des Programms noch gesetzt werden können. Somit kann es von zwei bis vier Spielern gespielt oder die Spielfeldgröße auf 18x18 oder 20x20 festgelegt werden. Eine der größeren Auswirkung hat die Auswahl der Spielregeln. Connect6 ist ähnlich wie Vier gewinnt und Ziel ist es mehrere Steine in eine Reihe zu legen. Connect6 wird dabei aber auf einem Brett gespielt, das eben auf dem Boden liegt. So fallen die Steine nicht bis in die letzte Zeile sondern bleiben an ihrem Platz. Ebenso werden pro Zug zwei Steine pro Spieler gesetzt. Bei der Standart-Spielregel zählen die Steine nur bis zum Spielfeldrand. Es gibt aber auch die Torus-Version, bei dem die Steinreihen über den Rand hinaus gezählt werden. So beginnt das Spielfeld erneut von oben, wenn man aus der untersten Zeile eins nach unten gehen würde.

Zusätzlich dazu soll jetzt auch noch Vier Gewinnt mit unterschiedlichen Variationen/Regeln implementiert werden. Bei einer Variation soll es möglich sein, die Spielsteine nur von oben herunterfallen zu lassen. Bei einer anderen z.B. Drop Four ist es aber auch möglich den untersten Stein einer Spalte zu entfernen. Bei dieser Spielesammlung soll es möglich sein mit wenig Aufwand neue Spielregeln/Variationen oder sogar ganz neue Spiele (die einen ähnlichen Aufbau haben) zu implementieren.

Bei der Entwicklung dieser Software wurde zwar darauf geachtet einen möglichst strukturierten und einfachen Code zu erstellen, es wurde sich aber nicht direkt an die in dieser Vorlesung besprochenen Themen gehalten. Es wurde zwar versucht die DRY-Regel anzuwenden, da diese mir schon selber in den Sinn gekommen ist, aber wie ich später festgestellt habe wurde diese nicht überall umgesetzt. Ebenso habe ich öfters kleinere Code Reviews vorgenommen und refactored, da mir immer wieder beim erneuten Anschauen aufgefallen ist, wie unverständlich der Code ist. Da der Code ursprünglich von einer Programmieraufgabe am KIT stammt, mussten auch Kommentare verwendet werden. Nachdem ich diesen Code aber nach über 3 Jahren wieder angeschaut habe ist mir klar geworden, dass trotz vieler Kommentare der Code sehr schwierig zu lesen ist.

Beschreibung des Kundennutzens Der Kunde erhält Spaß und Unterhaltung.

Verwendete Technologien - Java - JUnit 5 - IntelliJ

Link zum Repository `https://github.com/a-schmider/ASE_Substrat`

2 Unit tests

3 Programming Principles

4 Clean Architecture

Dieses Programm wurde in zwei Schichten aufgeteilt. Dem Kern-Modul, in dem die ganzen Strukturen liegen, und dem Plugin-Modul, in dem momentan die Ausgabe/-GUI realisiert wird. Um sicherzustellen, dass die Klassen im Kern die Plugins nicht kennen wurden Java Module verwendet und es wurde nur dem Plugin Modul gestattet die Klassen des Kerns zu verwenden und nicht andersherum. Dies ist wichtig, damit die Plugins ausgetauscht werden können ohne den Domain Code anpassen zu müssen.

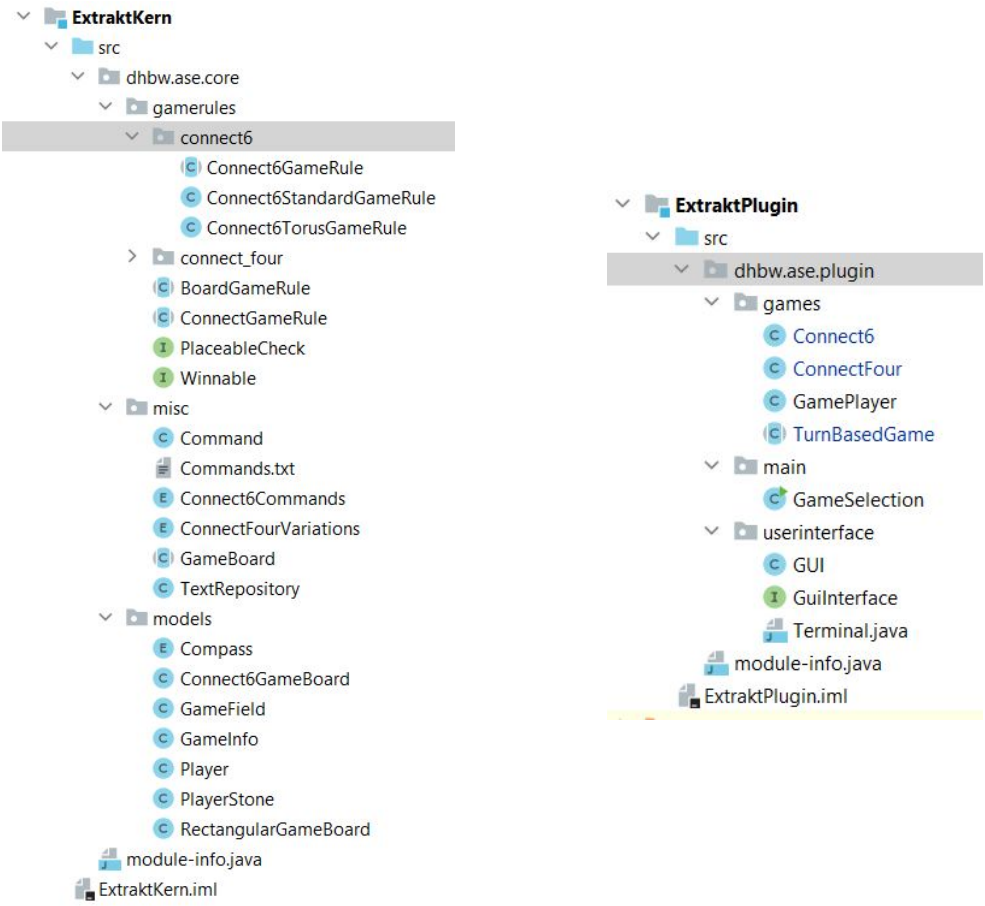

Abbildung 1: Kern- und Plugin-Module

5 Entwurfsmuster

6 Refactoring