

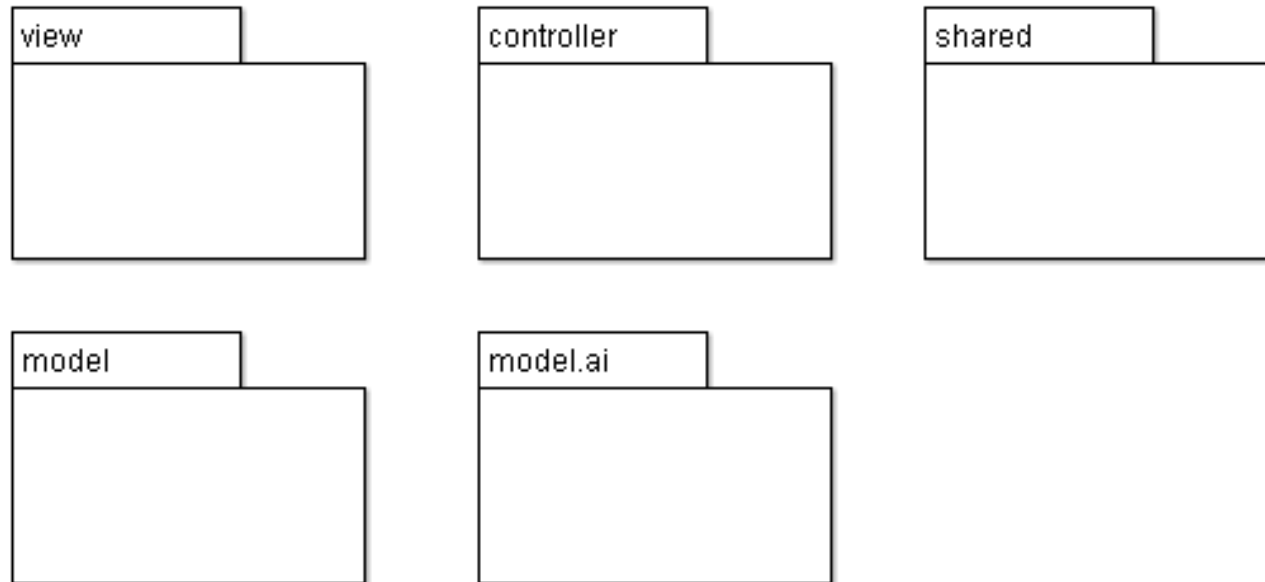
4 Gewinnt – Überblick



Übersicht

Thema	Vier Gewinnt
Gruppe	4 – „Peach“
Mitglieder	Martin Kapfhammer Benjamin Weller Benedikt Weber Tim Karl Eike Hildebrandt
Studiengang	WWI08C – 5. Semester
Version	1.0
Stand	13.10.2010
Autoren	Martin Kapfhammer

Packages



Packages

▶ Controller

- Definieren den Ablauf des Spiels
- GameController
 - Ablauf eines Spiels, ab wann ist es zuende usw
- Actions
 - Implementieren das ActionListener Interface, reagieren auf Buttons Clicks etc.
- Main
 - Startet das Game

Packages

▶ Model

- Hier befindet sich die Businesslogik, z.B. eine Klasse, die bestimmt, ob ein Spieler gewinnt, wie ein Spiel abläuft etc.
- Auch die Klassen zum schreiben in Datenbanken befinden sich hier

▶ Model.Ai

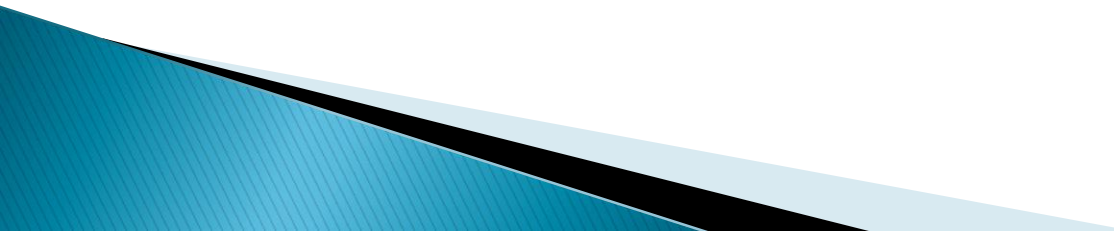
- Alle Klassen, die eine Künstliche Intelligenz implementieren
- Klassen, die eigentlich keine KI implementieren, aber das AiInterface, und die Zugdaten z.B. aus einer XML-Datei auslesen

Packages

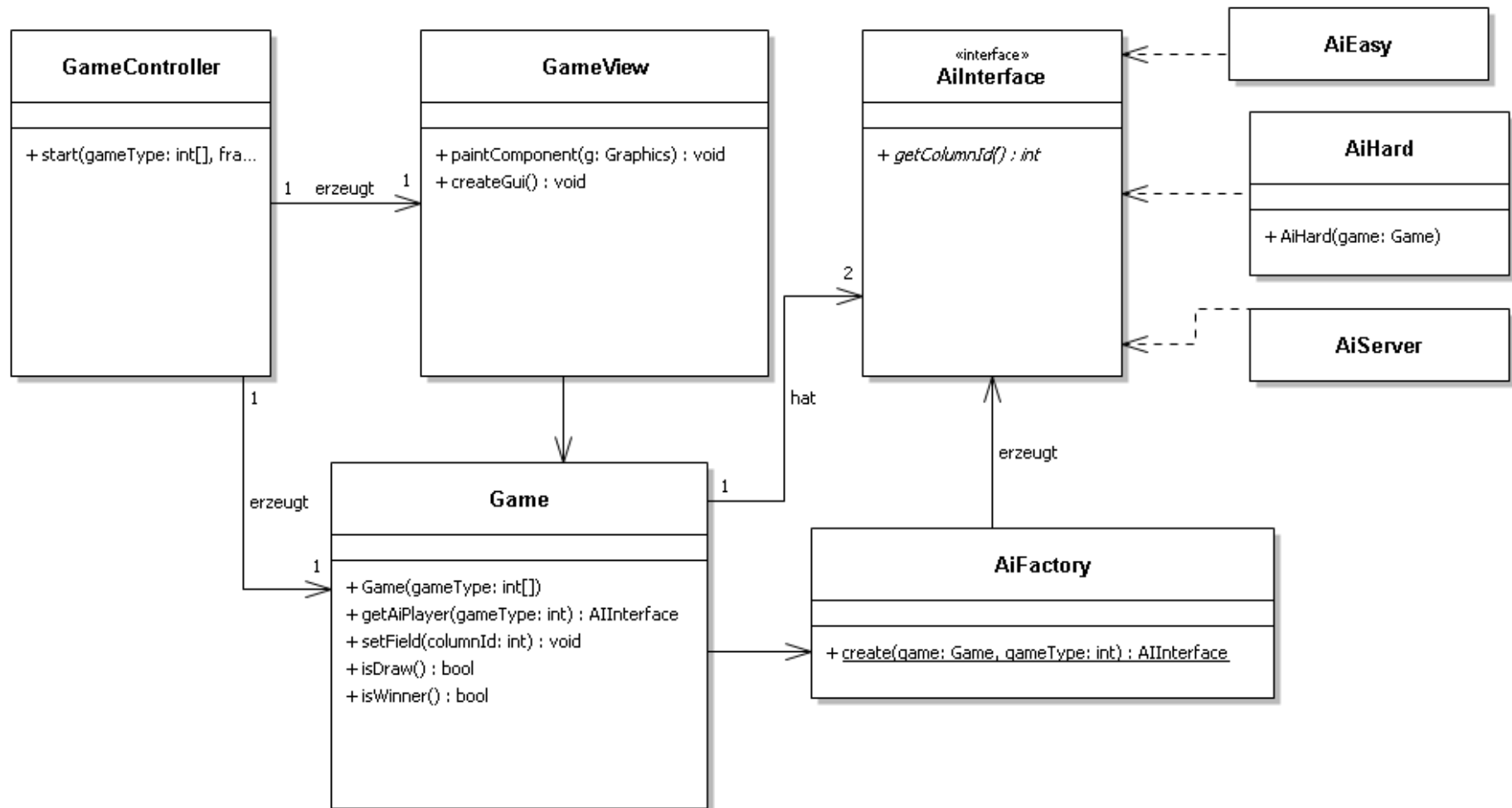
▶ Shared

- Hier befinden sich Klassen, die in allen Teilen der Anwendung benutzt werden, z.B. Globale Konstanten

▶ View

- Alle Klassen, die GUI-Elemente enthalten
 - MainFrame: Das Frame mit Menüleiste
 - Eine weitere View wird in der MainFrame angezeigt
 - Komplexe Windows/PopUps
- 

Klassendiagramm – Auszug



Klassendiagramm zwischen Analyse und Design
– Ablauf eines 4-Gewinnt Spiels

Erläuterung

▶ GameController

- Der GameController startet ein neues Spiel
- Speichert, wer wie viele Sätze gewonnen hat
- Entscheidet, ob Spiel zu ende ist
- Bei einem Start wird eine neue GameView erzeugt und auf den MainFrame hinzugefügt
- Außerdem wird eine neue Game-Logik Klasse erzeugt

Erläuterung

▶ GameView

- Hier wird die Gui erzeugt
- Zum einen der Statistik-Block, bestehend aus JLabels und JPanels
- Zum anderen das Spielfeld, das gezeichnet wird
- Auch auf Events wird reagiert
- Bei Klick auf ein Feld wird die setField-Methode des Game-Objekts aufgerufen, und das Spielfeld neu gezeichnet

Erläuterung

▶ Game

- Enthält die Logik
- Das Feld als 2D-Array
- Überprüft ob aktueller Spieler gewonnen hat
- Oder ob alle Felder besetzt
- Erzeugt außerdem die Computergegner, über die AiFactory, und kann diese zurückliefern

Erläuterung

▶ AiFactory

- Statische Fabrik Methode, die je nach GameType eine Ki erzeugt
- Handelt es sich um einen Nutzer, der spielt, dann wird *null* zurückgegeben

▶ AiInterface

- Interface für die KI Implementierungen

Einstellungen

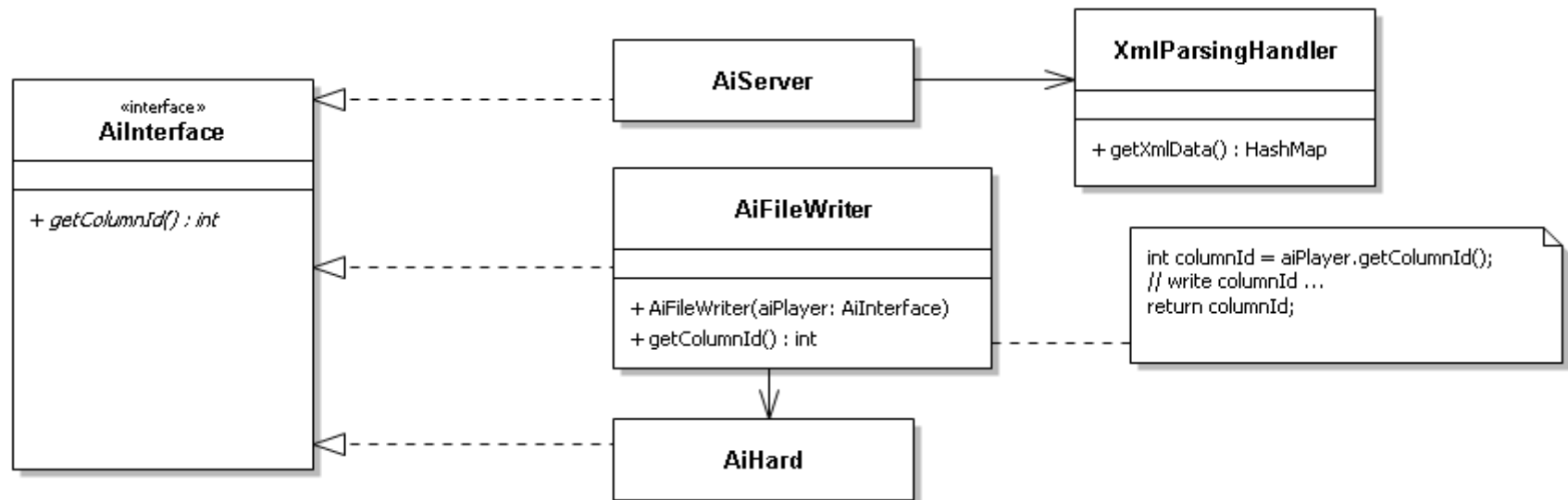
Preferences
- names: String[] - gameType: int[]
+ <u>getInstance() : Preferences</u>

- ▶ Singleton-Implementierung, siehe:
http://de.wikipedia.org/wiki/Singleton_%28Entwurfsmuster%29
- ▶ Dadurch wird gewährleistet, dass während eines Spiels immer die selben Einstellungen verwendet werden
- ▶ Hier werden die Nicknames der beiden Spieler, sowie der gewählte GameType gespeichert
- ▶ Die Einstellungen werden im Einstellungspopup gemacht

DHBW – Turnier

- ▶ Zum Auslesen des Gegnerzugs: AiServer, der keine KI implementiert, sondern den Zug aus der XML-Datei vom Server ausliest.
- ▶ Zum Setzen des eigenen Zugs: Es wird eine KI verwendet (AiHard), der mit Hilfe des Decorator Pattern (<http://de.wikipedia.org/wiki/Decorator>) über die Klasse AiFileWriter das Verhalten hinzugefügt wird, die gefundene Spalte in eine Datei zu schreiben.

Klassendiagramm – DHBW Turnier



Weitere Infos

- ▶ Weitere Informationen im JavaDoc, siehe <http://code.google.com/p/ebmb-cf/downloads/detail?name=Javadoc%20Stand%20220910.rar&can=2&q=>