**PROGRAMMING PROJECT**

SOFTWARE SYSTEMS

Lieke Hamelers (s1741640)

Amber Altenburg (s1477226)

Januari 2016

Universiteit Twente

Inhoudsopgave

[1 Discussion of the Overall Design 3](#_Toc471902457)

[1.1 Class Diagrams 4](#_Toc471902458)

[1.2 Requirements 5](#_Toc471902459)

[1.2.1 Requirements voor de server 5](#_Toc471902460)

[1.3 Model-View-Controller pattern 6](#_Toc471902461)

[1.3.1 Model 6](#_Toc471902462)

[1.3.2 View 6](#_Toc471902463)

[1.3.3 Controller 6](#_Toc471902464)

[1.4 Data Storage and Communication 7](#_Toc471902465)

[2 Discussion per Class 8](#_Toc471902466)

# 1 Discussion of the Overall Design

## 1.1 Class Diagrams

## 1.2 Functional Requirements

### 1.2.1 Functional requirements voor de server

|  |  |
| --- | --- |
| **Functional requirement** | **Geïmplementeerd door:** |
| 1. Wanneer de server wordt gestart, moet er een poort nummer ingevuld worden waarnaar de server zal luisteren. |  |
| 1. Als het poortnummer al in gebruik is, wordt er een passende foutmelding teruggegeven en kan er een nieuw poortnummer worden ingevoerd. |  |
| 1. Een server moet meerdere exemplaren van het spel, die tegelijkertijd door verschillende clients worden gespeeld, kunnen ondersteunen. |  |
| 1. De server heeft een TUI die ervoor zorgt dat alle communicatie berichten naar *System.out* worden geschreven. |  |
| 1. De server moet het protocol zoals gedefinieerd voor de tutorial groep tijdens de projectsessie in week 7 respecteren, dat wil zeggen, de server moet kunnen communiceren met alle andere cliënten uit de tutorial groep. |  |

## 1.3 Model-View-Controller pattern

### 1.3.1 Model

De volgende classes vormen het model: Board, Color, ComputerPlayer, Field, HumanPlayer en Player.

De model classes representeren de gegevens van het Connect Four 3D programma. Zij beheren de gegevens en voeren alle transformaties van die gegevens uit. Verder zijn deze classes niet op de hoogte van de controllers en views; ze bevatten geen verwijzingen naar beiden.

### 1.3.2 View

De volgende class vormt de view: TUIView.

De bovenstaande class beheert de visuele weergave van de gegevens die door het model worden gerepresenteerd. TUIView zorgt ervoor dat er data zichtbaar wordt voor de spelers van het Connect Four 3D programma.

### 1.3.3 Controller

De volgende classes vormen de controller: Game en ThreeDRow.

De controller classes combineren de gegevens uit het model met de view. Daarnaast reageren de controllers op gebeurtenissen die meestal het gevolg zijn van handelingen van een gebruiker, bijvoorbeeld het doen van een zet.

## 1.4 Data Storage and Communication

# 2 Discussion per Class

## 2.1 Board

Board maakt gebruik van de classes: Color en Field.

Zoals de naam al doet vermoeden, vervult de Board class de rol van het speelbord van Connect Four 3D. Board is dus verantwoordelijk voor de representatie van het speelbord. Board zorgt ervoor dat er een bord wordt gecreëerd met een grootte van 4 bij 4 bij 4. In eerste instantie zijn alle velden op het bord leeg. Alle velden op het bord hebben een x-, een y- en een z-coördinaat.

Naast de representatie van het speelbord is Board ook verantwoordelijk voor het controleren van een zet (oftewel: controleren of een zet een geldige zet is).

## 2.2 Game

Game maakt gebruik van de classes: Player, Board, Player en Color.

The class Game is verantwoordelijk voor de handhaving van een Connect Four 3D spel. Game doet dit door bij te houden welke spelers er meedoen aan het spel. Verder houdt Game ook bij welke speler aan zet is.