Programmeer Project
Vier Op Een Rij

Gerwin Puttenstein
s1487779

1 februari 2015

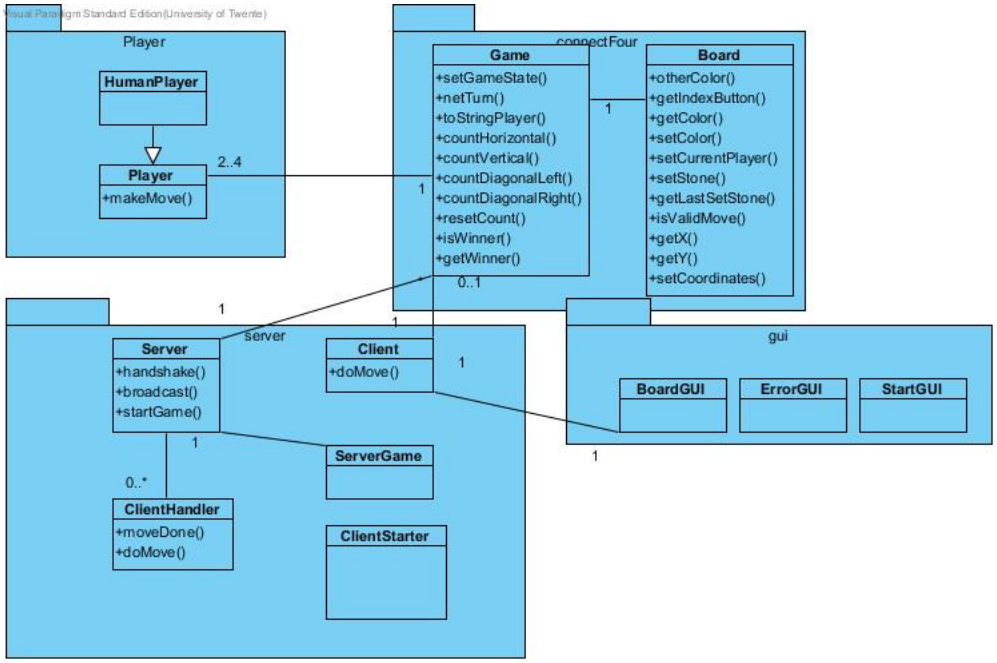
Inhoudsopgave

1 Inleiding

Een van de doelen van de afgelopen module, module Software Systemen, was het maken van het project. Als project wordt er een spel gemaakt. Dat spel moet over een server gespeeld kunnen worden en het spel moet zelf controleren of de spelers zich aan de regels houden en of er iemand gewonnen heeft als er een set gedaan is. Het gaat om een bordspel waarbij het niet gaat om geluk maar om strategie zodat er ook een AI voor gemaakt kan worden die door zijn strategie bijna standaard wint.
Dit jaar is het spel dat geïmplementeerd zal worden 'vier op een rij'. Dit is een spel voor twee personen. Bij 'vier op een rij' is het de bedoeling dat men stenen in een van de zeven rechtopstaande kokers gooit, wanneer een kolom vol is mag hier natuurlijk geen steen meer aan toe gevoegd worden. De spelers doen dit om de beurt en proberen er voor te zorgen dat de andere speler geen vier stenen, horizontaal, verticaal of schuin aansluitend heeft liggen en probeert tegelijkertijd dit wel zelf voor elkaar te krijgen. De eerste die dit lukt heeft gewonnen.
Als begin positie is het bord leeg. De persoon die begint is vrij om een rij te kiezen. Ook in elke volgende set is er geen beperking bij het kiezen van de kolom. De rij waarop de steen terecht komt is afhankelijk van het aantal stenen dat er al in die kolom liggen, alle stenen zullen namelijk naar beneden vallen. In het echte spel mag men onderling bepalen wie de eerste set mag doen, in mijn versie van het spel wordt dit door de volgorde waarop de spelers op 'play' klikken.

2 Bespreking van het Totale Ontwerp

2.1 Klassediagrammen



2.2 Vereisten

Er zijn een aantal dingen die wel gelukt zijn om te implementeren en een aantal dingen zijn helaas niet in het eindsysteem gekomen.
Voor de server zijn de volgende dingen in het systeem gekomen, ik maak gebruik van de lijst op pagina 18 als referentie:

1. Het poortnummer kan worden mee gegeven aan de server via de runconfiguratie in Eclipse.

2. Er wordt een error gegeven als er geen Server gestart kan worden op de gegeven poortnummer.

3. Er kunnen meerdere instanties tegelijkertijd gespeelt worden op de server. Deze houd namelijk een lijst bij van alles ServerGames die draaien. En deze ServerGames houden dan de spelers bij die in het spel zitten

4. Alle berichten worden geschreven naar System.out door een functie print binnen de Server klasse.

5. De server respecteerd het protocol dat we hebben opgesteld. Echter zijn niet alle commandos geïmplementeerd.

Voor de client zijn dat:

1. De client heeft een tui waarop hij het een en ander kan invoeren aan commandos. Wanner hij in een spel is kan hij gebruik maken van een simpele maar overzichtelijke gui.

2. De client heeft alleen support voor HumanPlayers, de AI is niet geïmplementeerd.

3. Omdat er geen AI is, kan je ook niet de denktijd van deze nietbestaande AI instellen.

4. De client heeft geen hint mogelijkheid. Deze zou kunnen worden gemaakt met behulp van een AI, maar deze is er dus niet.

5. Als een game voorbij is, komt de speler weer in de lobby. Vanaf daar kan hij opnieuw een spel starten.

6. Als een client crashed, wordt de client keurig gedisconnect van de server. Echter staat hij nog wel in de Game, als hij daar in was. Hier wordt hij nog niet keurig uit gedisconnect.

7. Als een server disconnect, worden alle verbindingen gesloten en zullen alle clients ook terminaten.

8. De client respecteerd het protocol. Echter is deze niet volledig geïmplementeerd. De meeste basic dingen zitten erin, zoals de hello en het maken van een move. Ook kan een client chatten.

2.3 Observer en Model-View-Controller

De BoardGUI is de observer en de Game klasse is observable. De BoardGUI is dan ook de view terwijl de Game het model is. De controller staat in de BoardGUI klasse als een klasse die BoardFrameController heet. Deze controller zorgt dat de communicatie tussen de BoardGUI en de Game goed verloopt. De Board FrameController merkt wanneer er een knop ingedrukt wordt en geeft dit door aan Game zodat die kan controleren of de set geldig is en of er na deze set een winnaar is of dat de volgende persoon gewoon mag spelen. Wanneer de set geldig is, dan wordt dit vakje geüpdate en zal de gui dit door krijgen en dit vakje van kleur laten veranderen.

2.4 Formats voor Data-opslag en Communicatie Protocollen

Welke vakjes gekleurd zijn houdt de BoardGUI bij. Deze heeft een lijst van alle vlakjes die of EMPTY, zwart, zijn of een kleur hebben omdat er een steen ligt. Ook de spelers zitten in een lijst. Op die manier kun je gemakkelijk switchten tussen de ene en de andere speler.
We hebben gekozen voor één communicatie protocol. In dit protocol is onderscheid gemaakt tussen commands voor de sever, de client, gedeelde commands en error meldingen die ontvangen en verzonden kunnen worden.

3 Beschrijving per Klasse

Hier zal van elke klasse kort beschreven worden waar deze verantwoordelijk voor is in het spel, wat de rol is van de klasse, wat de andere klasses zijn die zorgen dat deze klasse zijn verantwoordelijkheden kan uitvoeren, of er speciale gevallen zijn die omschreven zijn in het contract van de klasse en of er bepaalde voorzorgsmaatregelen zijn genomen om de randvoorwaarden te vervullen van de server klasse.

3.1 connectFour

In deze package zitten 'Board' en 'Game' Deze klasses zijn beide verantwoordelijk voor de regels van het spel en het controleren hier van.

3.1.1 Board

De rol van de klasse Board binnen het systeem is het zorgen dat de regels van het spel worden nageleefd, zo moeten de stenen naar beneden vallen tot waar ze kunnen, wanneer er een kolom wordt gekozen waarin deze steen moet komen te liggen. Dit zijn ook meteen de verantwoordelijkheden van de klasse Board. De klasse is bijvoorbeeld ook verantwoordelijk voor het controleren of het spelbord vol is en of er een winnaar is.
Om te zorgen dat Board kan doen wat hij moet doen zijn de enum 'PlayerColor' en 'GameState' vooral heel belangrijk. Deze zorgen dat de kleuren kloppen en dat het spel een status kan hebben waarin het bord zich kan bevinden.
Wat zeer belangrijk is in deze klasse, is dat de coördinaten die worden mee gegeven, dat die kloppen. Zo mag er geen kolom worden gegeven met een grotere waarde dan zeven en een lagere waarde dan nul. Ook de waarde van de rijen moet tussen de nul en de zes blijven.

3.1.2 Game

Game is het model van de MVC. Deze telt het aantal stenen dat er vanuit de net gelegde steen alle kanten op liggen om te zien of er misschien ergens vier of meer stenen van de zelfde kleur op een lijn liggen. Ook houdt game bij welke speler er nu aan de beurt is, in welke staat het spel nu is en als er een winnaar is, wie dit dan is.
Ook in deze klasse is 'GameState' dus blangrijk. Aangezien Game het model van de MVC is, is Game dus verantwoordelijk voor de communicatie met de GUI. Ook is Game de Obserable klasse. Dat betekend dat Game een spel kan starten door een bord en spelers aan te maken.

3.2 gui

In de package 'gui' zitten de klasses die verantwoordelijk zijn voor de graphic user interfaces. Hiervan zijn er drie, één voor het maken van het spelbord, één voor het maken van het start scherm en één voor het maken van het error scherm wat op komt als er iets fout gaat.

3.2.1 BoardGUI

De GUI is verantwoordelijk voor de grafische interface van het spel. Deze zorgt dus voor eer raster met zeven kolommen en zes rijen. Boven deze rijen is nog een extra rij met zeven knoppen. Om een steen in een kolom te laten vallen met je op de boven gelegen knop klikken.
De GUI is de view van de MVC. Deze communiceert dus met Game. De GUI geeft door welke rij er gekozen is, de Game en Board kijken dan of de zet mag en laten de steen in de gewenste kolom naar beneden vallen tot deze op de bodem, of de bovenste steen ligt.

3.2.2 BoardFrameController

BoardFrameController is een klasse binnen de klasse BoardGUI. Deze klasse is de controller van de MVC. Deze klasse krijgt van de BoardGUI door welke knop er ingedrukt is en geeft dit weer door aan de Client. De Client moet dan namelijk een bericht verzenden omdat deze een move gedaan heeft en geeft ook het index nummer mee van de knop die aan geklikt is.

3.2.3 ErrorGUI

ErrorGUI is een simpele klasse, deze klasse maakt een klein scherm me daarop de informatie die is mee gestuurd met het aanroepen van de klasse. Deze klasse kan dus gebruik worden voor het afhandelen van excepties die optreden tijdens het spelen van het spel. De errorGUI wordt ook verder nergens gebruikt. Hier waren nog plannen mee.

3.2.4 StartGUI

De functie van de StartGUI is het verzamelen van de informatie van de speler, zoals de naam en de poort waarover de speler wil spelen en het ip adres waarmee de speler wil verbinden. Ook kan de Speler hier een kleur kiezen waarmee hij wil spelen. Wanneer al deze informatie is mee gegeven, kan men op start klikken waardoor de BoardGUI zal worden aangeroepen en de spelers kunnen gaan spelen. Helaas is deze StartGui niet geïmplementeerd alleen gemaakt.

3.3 players

Player is een package met verschillende spelers die het spel kunnen spelen. Deze package bevat een Player, dit is een abstracte klasse en een HumanPlayer waarmee men het spel zelf kan spelen.

3.3.1 HumanPlayer

HumanPlayer is de AI loze speler die de persoon volledig toestaat om alles te doen wat hij of zij wil. HumanPlayer extends de Player klasse. Vanuit de StartGUI worden aan de player een naam en een kleur mee gegeven. Deze zijn in de StartGUI opgegeven door de persoon die het spel gaat spelen.

3.3.2 Player

De abstracte klasse Player maakt de speler aan, deze speler is bijvoorbeeld een ComputerPlayer of een HumanPlayer. De Player doet de set op het bord en geeft deze door aan Board. De klasse Player gebruikt de enum PlayerColor voor de mogelijke kleuren van de speler.

3.4 server

In deze package staan alle klasses die met het server gedeelte van het spel te maken hebben, zoals de Client en de ClientHandler

3.4.1 Client

De Client wordt aangemaakt wanneer er een speler een spel start. De Client kan een bericht versturen naar andere Clients via zijn client handler. De client schrijft de berichten die hij binnen krijgt op de System.in naar de socket.out. Dan handeld de clienthandler verder de berichten af en zorgt hij voor de communicatie tussen de client en de server. Ook reageerd de client op bepaalde berichten van de server. Zoals het starten van een gui en een board wordt hiermee geregeld.

3.4.2 ClientHandler

De ClientHandler wordt aangemaakt door de server zodra een Client zich meldt. De ClientHandler is een onderdeel van de server en zorgt voor de communicatie met de Client. Een ClientHandler neemt deze taak over van de server zodat de server een spel kan starten mocht dat nodig zijn en zich bezig kan houden met het bijhouden van de Clients en hun ClientHandlers. Elke Client heeft zijn eigen ClientHandler.

3.4.3 ClientStarter

Het was de bedoeling om deze klasse te gebruiken om een nieuwe client te kunnen starten met gegeven input. Echter is dit niet verder van de grond gekomen en niet geïmplementeerd.

3.4.4 Server

De Server zorgt dat er communicatie kan bestaan tussen de verschillende spelers. De spelers zijn de Clients. Wanneer een nieuwe Client zich meldt, maakt de server een ClientHandler aan. Deze ClientHandler zorgt vervolgens voor de communicatie. Wanneer er een bericht naar alle Clients gestuurd moet worden moet dit via de Server gaan omdat die een lijst bij houd met alle ClientHandlers en dus gemakkelijk een bericht naar alle Clients kan sturen via hun ClientHandler.
Ook start de Server een nieuwe game wanneer twee Clients daar om vragen en houdt hij bij of de regels van het spel worden na geleefd door de spelers.

3.5 tests

In dit package staan alle tests waarmee de verschillende klasses van het spel getest zijn.

3.5.1 GameTest

Deze klasse heb ik gebruikt om de gameregels te testen tijdens de ontwikkeling ervan. Het is niet een correcte test volgens de standaards, maar het voldeed voor toen.

3.6 utils

In de package utils staan de de GameState, PlayerColor en het ServerProtocol.

3.6.1 GameState

GameState is een enum die de status van de game definieert. Dit zijn FINISHED voor als een spel is afgelopen, INPROGRESS voor een spel dat bezig is, DRAW, voor als een spel als gelijk spel is geëindigd, dat wil zeggen dat het bord vol is, maar er geen winnaar is. Verder is er nog de NOTSTARTED, deze status komt eigenlijk nooit voor omdat dan de game meteen afgesloten wordt.

3.6.2 PlayerColor

PlayerColor is de enum die de kleuren van de spelers definieert.

3.6.3 ServerProtocol

ServerProtocol is een interface, in deze interface staat het protocol uitgeschreven waarnaar de server handelt. In de interface is onderscheid gemaakt tussen commands voor de server en de client, commands die alleen voor de client zijn, commands alleen voor de server en error berichten die verstuurd en ontvangen kunnen worden.

4 Test Verslag

Ik heb maar é én klasse voor het testen van mijn code, ik heb geen tijd gehad om meer testklasses te maken. Ook deze test klasse is een handmatige test, de klasse maakt een GUI aan waarin stenen geplaatst worden om te zien of de verschillende count methodes in Board werken. In de console wordt geprint of er een winnaar is en waar de steen geplaatst wordt en hoeveel van de zelfde kleur er in een richting liggen volgens elk van deze methodes. Verder heb ik het spel uirgebreid handmatig getest om te kijken of er dingen fout gaan wanneer er op bepaalde plekken stenen terechtkomen. Wanneer ik genoeg tijd had gehad had ik nog klasses kunnen maken die geheel zelfstandig de verschillende klasses testen. In dat geval zou er alleen maar geprint worden waar waardes niet gelijk zijn aan wat ze zouden moeten zijn en wanneer alles goed is zou dit geprint kunnen worden. Dit is een veel snellere manier on te controlleren of elke klasse naar behoren werkt. Over GameTest.java: GameTest is mijn enige test klasse. Deze klasse test de countfuncties omdat hier nog dingen fout gingen in mijn implementatie. De klasse start een GUI op en plaatst hier een aantal stenen in om te kijken of de count functies werken. Tijdens het runnen van de test wordt er in de console geprint wie de set gedaan heeft, hoeveel blokken er aanliggend zijn aan dje kleur volgens elk van de count methodes en of er een winnaar is. De klasse kijkt dus niet zelfstandig of elke waarde uit komt waar deze uit moet komen. Dit moet nog met de hand gedaan worden. Deze klasse is dus verantwoordelijk voor het testen van de count methodes. Voor dit testen heeft deze klasse natuurlijk de klasse Game nodig omdat deze klasse de klasse Game test. Verder heeft hij ook de klasse BoardGUI nodig omdat deze klasse via de GUI test of de functionaliteit van de count methodes werkt.

5 metrics Rapport

De MLoC is een van de metrics die we gebruikt hebben. Deze is afhankelijk van hoe de code is form gegeven. Zo krijgen methodes andere waardes door het toevoegen of weghalen van witregels. De McCabe Cyclomatic Complexity geeft de complexheid van een methode weer. Deze kun je ook uittekenen, het gaat hier dan om het aantal forloops dat gebruikt wordt. De complexheid van een methode kun je dan bijvoorbeeld berekenen aan de hand van het aantal lijnen en punten dat gebruikt wordt. LCOM geeft een waarde voor het gebrek aan samenhang in methodes.
Bij het programmeren van mijn code heb ik erop gelet dat de ik niet onnodig lange methodes heb gemaakt en dat ik niet onnodig veer forloops in elkaar heb gestopt. Hierdoor zal de qualitiet van mijn code vrij goed zijn.

6 Reflectie op Planning

6.1 Ervaringen van week vier

In de vierde week van de module zijn we in het design project gaan kijken naar wat er zoal gedaan moest worden en hoeveel tijd de losse onderdelen zouden gaan kosten. Ook hebben we gekeken naar welke dingen eerst af moesten voor we verder konden. Dit is mij erg van pas gekomen bij het plannen van mijn programmeer project. Sommige dingen, zoals het uitbreiden van de GUI bijvoorbeeld had ik heeft veel zin in maar heb ik toch nog aan de kant geschoven om eerst de verplichtte vereiste af te krijgen.

6.2 Project planning

Het is mij niet goed gelukt om me aan mijn planning te houden. Dit kwam omdat er nogal veel tussendoor is gekomen zoals het leren voor de herkansingen en het afmaken van de opdrachten van de practica.
Vooral het server gedeelte van het project was veel meer werk dan gepland. Ik had verwacht dat dit minder werk was. Hier ben ik ook af en toe goed op blijven steken waardoor ik ook veel tijd ben kwijt geraakt. Toen ik hulp ging vragen hierbij ging het meteen al een stuk beter dus dat was wel heel fijn.

6.3 Compensatie voor de verloren tijd

Er was helaas niet veel tijd meer over om nog extra in het project te steken. Ik heb wel langere dagen gemaakt dan gepland en verder heb ik veel tijd die ik bijvoorbeeld wilde steken in het mooier maken van mij gui gestoken in het afmaken van de belangrijkere dingen zoals de server.
Ik denk nog steeds dat het project op zich was niet te groot om alleen te doen en af te krijgen, het waren meer de dingen die er tussen door kwamen die het zo druk maakte. Ik kan natuurlijk niet garanderen dat ik alles af had gekregen wanneer ik geen herkansingen tussendoor had gehad maar ik was in ieder geval een heel eind verder gekomen.

6.4 Wat ik geleerd heb

Wat ik geleerd heb, is vooral dat het heel behulpzaam kan zijn om mensen om hulp te vragen. Om dan maar, ondanks dat het daar druk is en ik me slechter kan concentreren, toch naar practicum te gaan en daar te vragen hoe ik nu verder moet in laatst van proberen het allemaal zelf op internet te vinden. Veel antwoorden

kun je op internet vinden maar helaas niet alles, daar heb je soms iemand voor nodig die kan helpen met het uitzoeken waar nou precies in mijn geval de fout zit.

6.5 Mijn advies

Als student assistent zou ik mijn studenten vooral adviseren op tijd te beginnen, dit heb ik wel gedaan maar ik weet dat veel mensen hier vaak de fout bij ingaan. Verder zou ik ze adviseren snel hulp te vragen als ze vast zitten. Het heeft geen zin om je hoofd te blijven breken over een bug die je niet ziet maar die een ander er binnen een paar minuten uit heeft. Ik zal ze ook vertellen dat ze het project vooral niet moeten onderschatten. Verder dat ze vooral ook niet naar hun scherm moeten blijven staren als het niet lukt. Als je er even niet uit komt is het het beste om even te gaan lopen en een kopje drinken te halen. Wanneer je dan terug komt zie je misschien wel meteen wat er aan de hand was en kun je weer verder.

7 Nawoord

Ondanks dat het nogal een uitdaging was het project af te krijgen heb ik geen spijt dat ik het project alleen heb gedaan. Helaas had ik in de kerst vakantie door het leren van mijn herkansing Discrete Wiskunde geen tijd gehad om aan deze module te werken. Hierdoor ben ik vooral in tijdnood gekomen waardoor ik helaas met minder plezier aan mijn project heb kunnen werken. Het bouwen aan een spel, hierom wilde ik ook graag alleen werken, zodat ik alles goed mee zou krijgen. Dat is ook aardig gelukt, ik heb veel geleerd van het bouwen van de server, iets wat ik waarschijnlijk niet had gedaan als ik met iemand samen had gewerkt.