

R0457098

# BALDER CROQUET REVERSI 2014-2015

Programmeren Visuele Gebruikersomgevingen

## INLEIDING

In dit project heb ik mijn interpretatie van het spel reversie gemaakt. Het is erg simplistisch en de klemtoon ligt meer op het vervangen van het originele spel op een aantrekkelijke manier dan een nieuw spel ontwikkelen. Ik ondersteun daarom geen afwijkingen op de originele spel regels.

## GRAFISCH OVERZICHT

### Toelichting

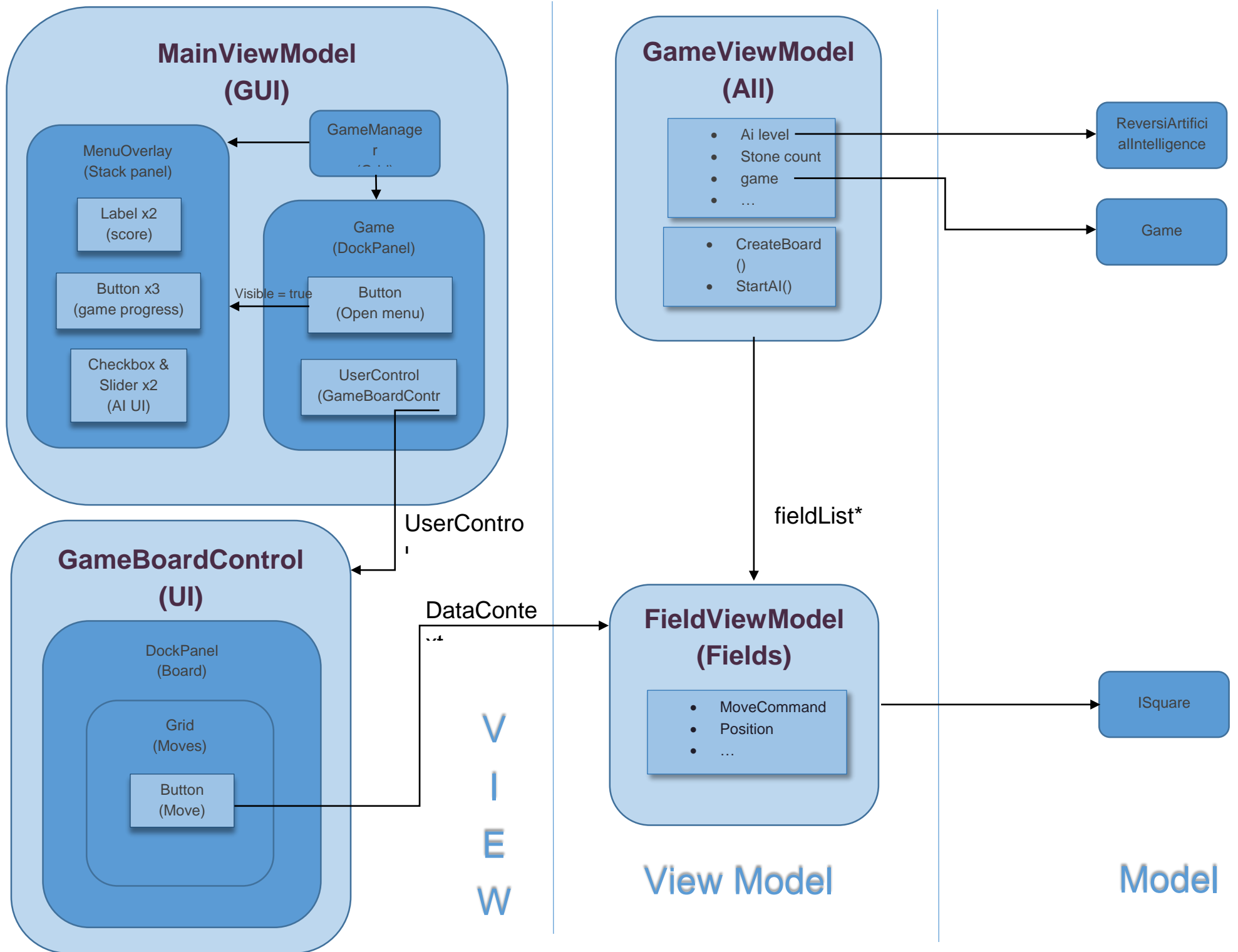
Om overzicht te houden werd er gebruik gemaakt van een MainView en nog een User Control genaamd GameBoard control. GameBoardControl is verantwoordelijk voor al de gamemechanics zoals; het aangeven van de volgende stap, de pionnen, ... terwijl in MainViewModel enkel de User Interface wordt behandeld; scores weergeven, menu openen, AI instellen...

Binnen de MainViewModel werken we met 2 grote panels, een is ons game board en de andere is onze menu overlay. Als er bij ons spel bord op Menu wordt gedrukt wordt dit menu geopend.

In de GameViewModel maak ik in de constructor een lijst met FieldViewModels aan, Deze krijgen bij aanmaak een ISquare en een position mee. Deze position kunnen we dan gebruiken om de fieldViewModels te linken aan de button Grid in de GameBoardControl. In e FieldViewModel wordt ook een click command gehandled die zorgt voor de speelbaarheid van het spel.



*Het grafisch overzicht bevat uiteraard niet alle elementen, enkel de elementen die gebruikt werden voor de algemene game flow worden besproken om het simpel en overzichtelijk te houden*



## Features

- Het spel kan gespeeld worden door 2 spelers tegen elkaar (zwart tegen wit)
  - Je kan zowel zwart als wit op automatisch laten spelen
  - Je kan ook het niveau van de AI instellen
- Het spel kan opnieuw gestart worden
- Het spel kan gestopt worden
- Het spel geeft aan wie er gewonnen heeft als er voor 1 van de 2 spelers geen zetten meer zijn en vraagt of je opnieuw wilt beginnen.
- De grote van de window staat vast om te garanderen dat de textures mooi uitkomen

## WPF Features

- Window resources(uiterlijk van buttons voor menu)
  - Controltemplate
  - Triggers (hover en klik)
  - Setters (borderbrush)
- Image source (score)
- Bindings (AI features)
- UpdateSourceTrigger (update als waarde van checkbox veranderd)
- IsEnabled (enable slider als AI checked is)
- Stackpanel met button en label (image button)
- LinearGradientBrush (menu balk)
- Button template (menu knop)
- IValue converter (current player => pion Texture)
- UserControl
- imageBrush (marble background)
- binding ICommand (button met ICommand verbinden; can execute en execute)
- SolidColorBrush

## BESLUIT

Ik heb me tijdens dit project voornamelijk gefocust op het uiterlijk om dat dit een van mijn zwakke punten is en omdat WPF het ideale framework is om dit een intuïtieve experimenterende manier bij te schaven. Mijn eigen doelstelling voor dit project was om een spel te creëren dat professioneel over kwam. Ik ben hier half in geslaagd; wat mij het meeste stoort is dat ik geen tijd meer had om multi threading te implementeren waardoor de zetten van de AI niet vloeiend over komen, ook past het simplistische menu niet erg goed bij het authentieke reversi bord.