

Startdocument voor de C#-2 eindopdracht

Startdocument van **Rick Vinke, Jesse Vugteveen, Jesse Oost, Max Appeldorn** en **Cedric Smit**.

Beschrijving

Voor de eindopdracht van C#-2 gaan we een collectie van kleine spellen maken. Deze collectie zal bestaan uit de volgende 3 spellen:

- Dammen
- Vier op een rij
- Memory

Een spel zal gestart kunnen worden van een start scherm.

Memory zal alleen te spelen zijn en het de tijd en aantal zetten bijhouden zodat je deze kan verbeteren. Vier op een rij en dammen zullen 'multiplayer' spellen worden die je tegen iemand anders kan spelen. Deze multiplayer spellen kunnen op 2 manieren worden gespeeld:

- Beide spelers op dezelfde computer laten spelen door afwisselend controle te geven aan de spelers.
- Met 2 computers via het lokale netwerk.

Voor de applicatie zal gebruik worden gemaakt van .NET 6 en .NET Maui met de MVVM-architectuur.

In- en uitvoer

In deze sectie wordt de in- en uitvoer van de application beschreven.

Invoer

In de onderstaande tabel wordt alle invoer (die de gebruiker moet invoeren om de toepassing te laten werken) beschreven.

Case	Type	Conditions
Player name	String	Kan niet null zijn.
Ip-adres	String	Moet een geldig IP of hostname zijn.

Uitvoer

In de onderstaande tabel wordt de verschillende uitvoer van de applicatie beschreven.

Case	Type	Conditions
Aantal beurten in Memory.	int	Kan nooit minder zijn dan het aantal memory kaarten / 2 .
Naam van de speler die aan de beurt is.	String	Kan niet null zijn.

Case	Type	Conditions
Naam van de speler die een spel heeft gewonnen.	String	Kan niet null zijn.
De index van de ingevoerde steen van vier op een rij.	int	
Of de geselecteerde Memory kaarten hetzelfde zijn.	boolean	
Het aantal dam stenen van een speler.	int	
De positie van de huidige dam steen.	int	

Calculaties

In de onderstaande tabel worden de calculaties van de applicatie beschreven.

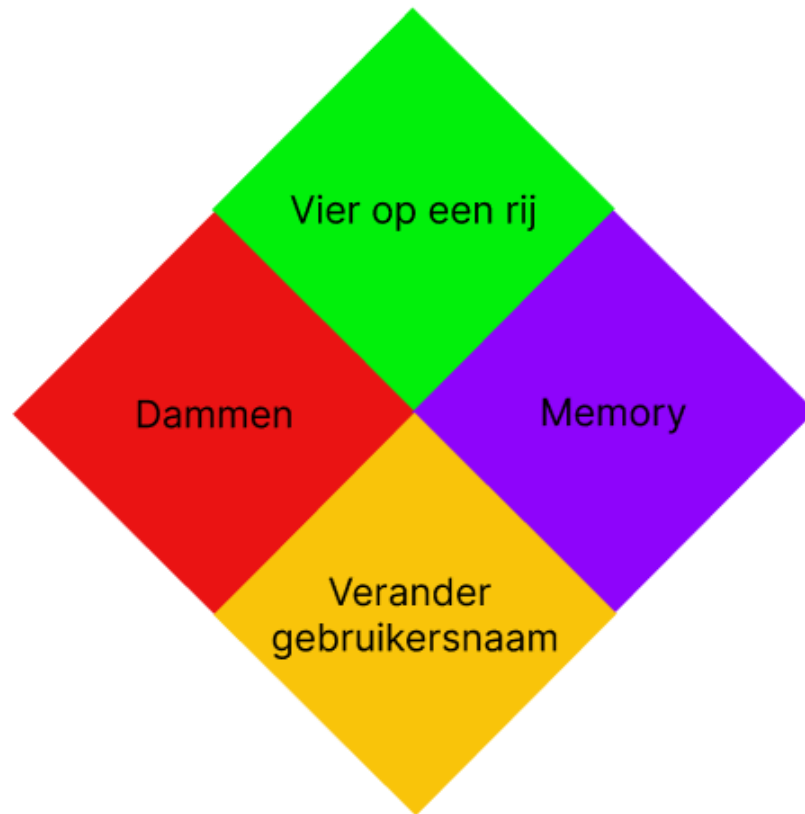
Case	Calculatie
Memory kaart vergelijking.	Calculeren of de twee geselecteerde Memory kaarten hetzelfde zijn.
Memory win status.	Calculeren of de Memory game is gewonnen.
Vier op een rij steen calculatie.	Calculeren of de vier op een rij steen kan vallen en hoe ver die kan vallen.
Vier op een rij win calculatie.	Calculeren of er vier op een rij is.
Dammen mogelijke zet calculatie.	Calculeren welke zetten de speler kan maken met dammen.
Dammen zet actie calculatie	Calculeren of de actie van de speler de steen van de andere speler pakt.
Dammen win status	Calculeren of een van de spelers dammen heeft gewonnen.
Netwerk gegevens uitwisselen	Stuur gegevens tussen de spelers zodat de gegevens gelijk zijn.

Klassendiagram

Deze volgt nog voor week 4.

Wireframes

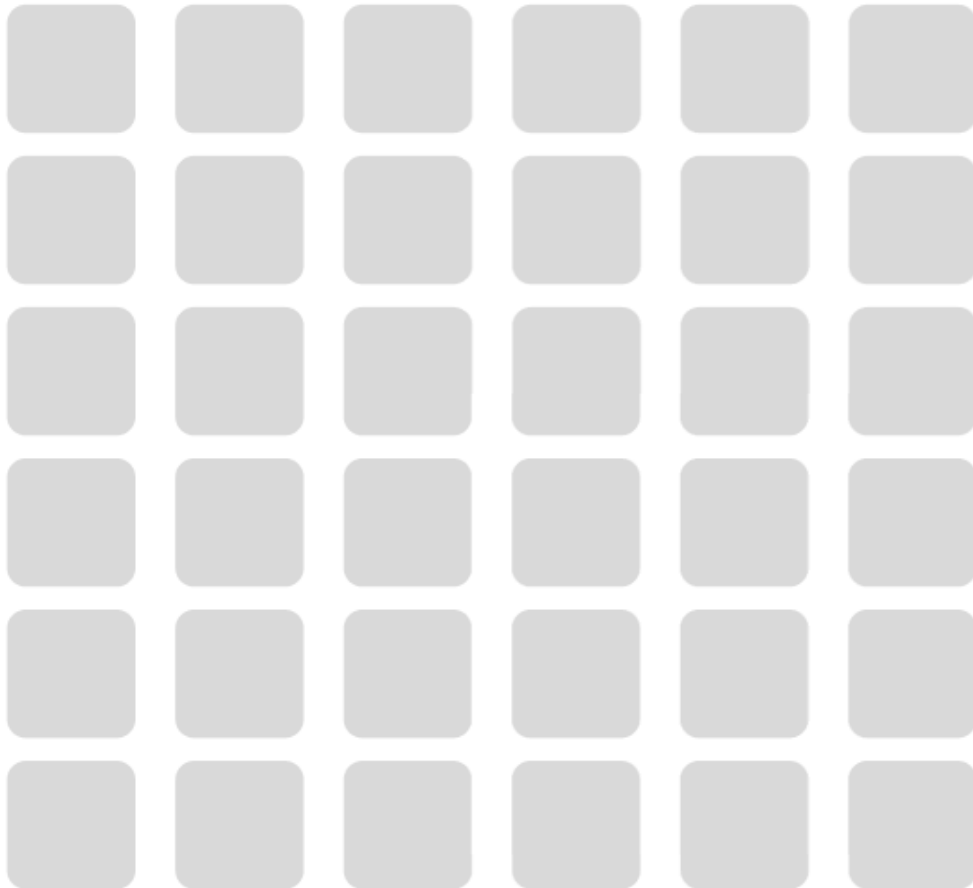
Menu scherm



Dit is het menuscherm waar de gebruiker een game kan kiezen, ook is er de mogelijkheid om de gebruikersnaam te veranderen

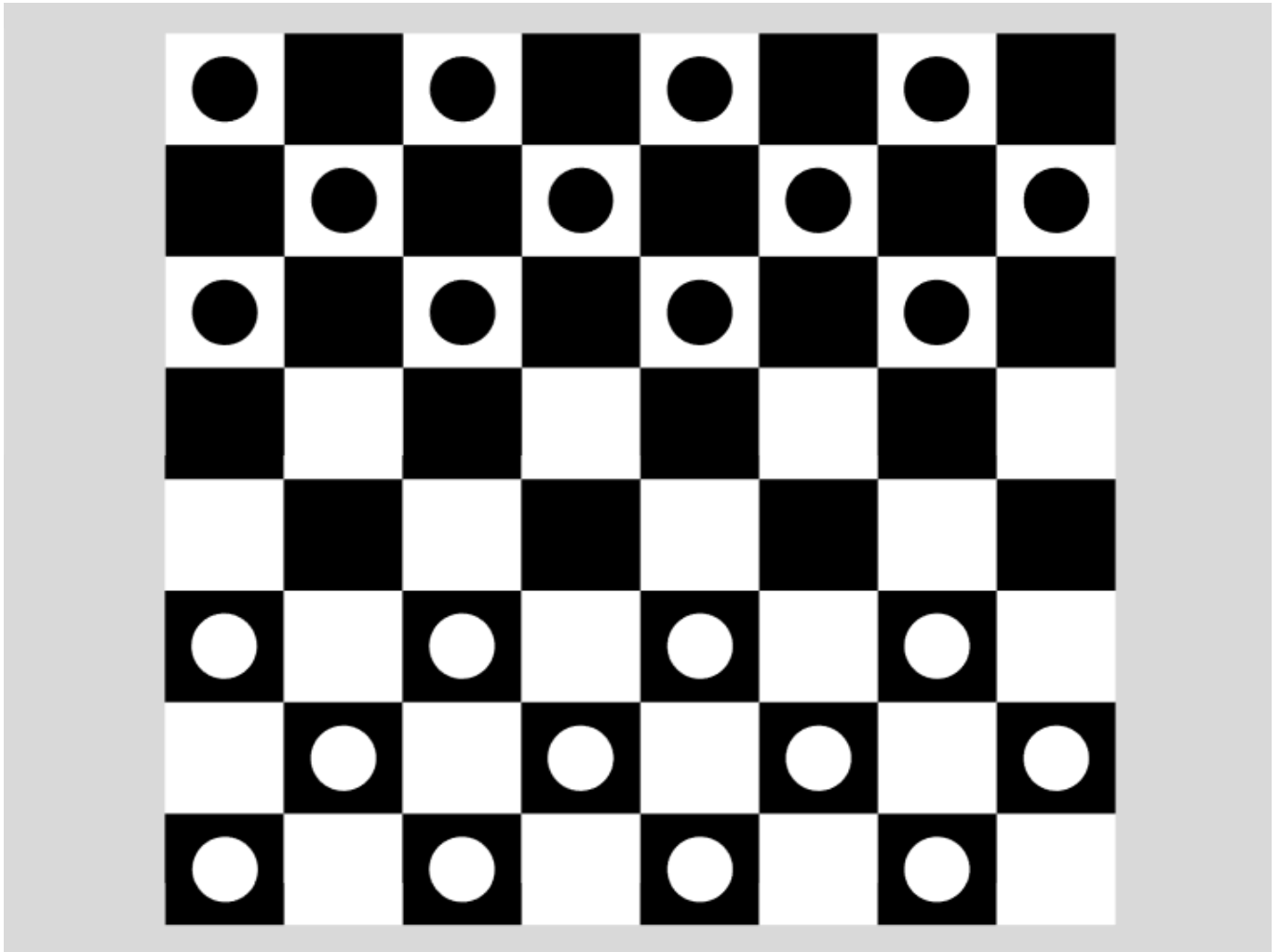
Memory

Aantal beurten:



De memory game. De gebruiker mag per keer twee kaartjes omdraaien. Als deze hetzelfde zijn worden de kaartjes uit het spel gehaald. Als de kaartjes niet hetzelfde zijn worden ze weer omgedraaid. Verder wordt bijgehouden hoeveel beurten de gebruiker nodig heeft om het spel te voltooien.

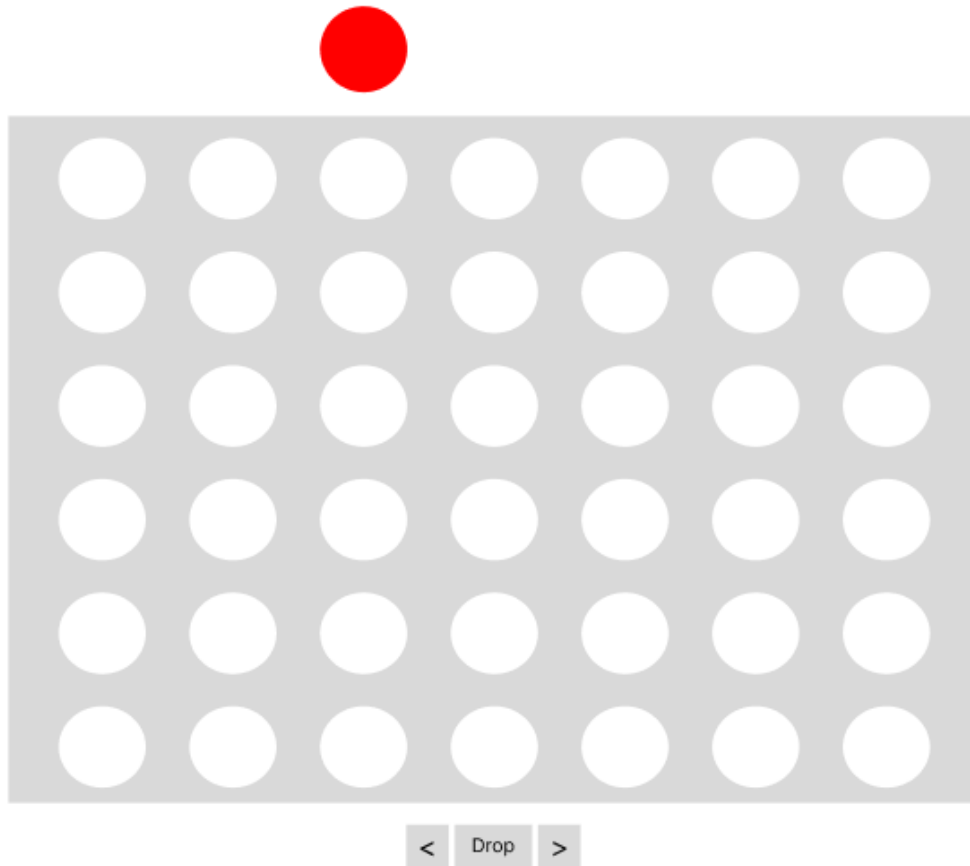
Dammen



Traditioneel dammen. Spelers kunnen via een lokaal netwerk tegen elkaar spelen. Degene die als eerste alle stukken van de tegenstander weg heeft gespeeld wint.

Vier op een rij

Speler {naam} is aan zet



Vier op een rij, twee spelers spelen tegen elkaar en mogen om de beurt een zet doen. Degene die als eerst horizontaal, verticaal of diagonaal vier op een rij heeft wint.

Planning

Week 1

- Het idee

Week 2

- Startdocument

Week 3

- Klassendiagram
- Requirements analyse
- Code conventies

Week 4

- Dashboard
- Tussenpeiling

Week 5

- Memory

Week 7

- Dammen
- 4 op een rij

Week 8

- Applicatie testen
- Eindpresentatie

Testplan

Het testplan is afhankelijk van dat de klassen bekend zijn. Dus deze komt samen met het klassendiagram voor week 4.