

Practicum 2: Vier op een rij

BVP 2013-2014

2 december 2014

In dit practicum ga je je eigen versie van het spel “Vier op een rij” ontwerpen. De details over het spel zelf vind je in sectie 1. In sectie 2 is een voorbeeld gegeven van hoe het spel er uit kan zien. Tenslotte krijg je in sectie 3 enkele suggesties over welke klassen je best gebruikt. Maar eerst enkele praktische afspraken:

- Deel 1.
Dit document is de opgave voor deel 1. Je zorgt ervoor dat dit deel af is voor je oefenzitting in week 51 (16 tem. 22 december). Dit zou je op minder dan 15 uur moeten kunnen maken.
- Deel 2.
Tijdens je oefenzitting in week 51 (16 tem. 22 december), krijg je de opgave voor deel 2 van het practicum. Deze opgave is een kleine aanpassing of uitbreiding van deel 1, en kan je maken in 2 uur. De opgave is anders voor elke reeks. Na deze 2 uur volgt de evaluatie. Tijdens de evaluatie zal een assistent naar je code kijken, en vragen stellen hierover. Dit duurt ongeveer 15 minuten.

1 Het Vier op een rij spel

In dit spel leggen 2 spelers om de beurt een schijfje met hun eigen kleur in één van de kolommen van een twee dimensionaal bord. Eén schijfje zal telkens de eerstvolgende lege plaats vanaf onderaan innemen. Het spel is afgelopen als één van de spelers vier op een rij heeft (en de respectievelijke speler daardoor wint), of als het bord vol is (en het gelijkspel is). Vier schijfjes op een rij kunnen zowel horizontaal, verticaal als diagonaal gevormd worden. In onze versie van het spel zal een speler het opnemen tegen een computerspeler (sectie 1.2.2).

1.1 Begin van het spel

In het begin van het spel kiest een speler de grootte van het spelbord, en de moeilijkheidsgraad van de computerspeler (makkelijk, normaal of moeilijk). Verder wordt

ook beslist hoeveel aambeelden elke spelers krijgt (1 of 2). Elke speler krijgt een gelijk aantal aambeelden in het begin van het spel.

1.2 Verloop van het spel

Het spel begint met een zet van de menselijke speler. Na een zet van beide spelers wordt het spelbord opnieuw afgedrukt.

Voorbeeld van een spelbord:

1	2	3	4	5	6	7	
_	_	_	_	_	_	_	Computer: 1 [#, #]
_	_	_	_	_	_	_	Stijn: 2 [#]
_	_	_	O	_	_	_	
_	_	O	X	_	O	X	
_	_	X	O	#	O	X	
_	X	X	O	O	X	O	
Computer (X) koos kolom: 2							

Het spelbord moet (minstens) de volgende elementen bevatten:

- spelbord met kolomnummers
- naam, score en aantal overgebleven aambeelden ([], [#] of [#,#]) van de spelers
- verschillende symbolen voor de schijfjes van de spelers (X voor computerspeler, O voor menselijke speler, # voor aambeelden)
- laatst gekozen kolom

1.2.1 Aambeeld

Het spel bevat één bijzonder schijfje, het “aambeeld”-schijfje (aangeduid met het symbool #), waarvan spelers maar een beperkt aantal (1 of 2) krijgen in het begin van het spel. Als een speler dit schijfje in een kolom steekt, zullen de x bovenste schijfjes die al in het spelbord zaten (ook evt. aambeelden van de andere speler) in

de respectievelijke kolom verwijderd worden. Hierna zal het aambeeld blijven liggen op de rest van de schijfjes in die kolom. Als x groter is dan het aantal schijfjes dat (voor het aambeeld gebruikt werd) in de kolom zat, zal het aambeeld eindigen op de onderste rij. Het aambeeld zal in het spelbord blijven voor de rest van het spel, tenzij het verwijderd wordt door een ander aambeeld. Een aambeeld kan niet gebruikt worden om 4 op een rij mee te vormen. In het begin kiest de speler het aantal aambeelden (1 of 2) dat elke speler (menselijk en computer) krijgt, en een waarde voor x .

1.2.2 Computerspeler

Het gedrag van de computerspeler zal voorgeprogrammeerd worden, afhankelijk van de gekozen moeilijkheidsgraad:

- *Makkelijk*
De computerspeler kiest een willekeurige kolom, en gebruikt geen aambeelden.
- *Normaal*
De computerspeler probeert elke mogelijke zet (een zet is een schijfje in één van de kolommen van het spelbord steken), en analyseert voor elke zet de uitkomst van het spelbord. Als er geen enkel spelbord 4 op een rij bevat, zal hij een willekeurige kolom kiezen (zoals in "makkelijk"). Ook bij deze moeilijkheidsgraad worden geen aambeelden gebruikt.
- *Moeilijk*
De computerspeler probeert elke mogelijke zet (een zet is een schijfje **OF** een **aambeeld** in één van de kolommen van het spelbord steken), en analyseert voor elke zet de uitkomst van het spelbord. Voor alle mogelijke uitkomsten van het spelbord wordt een score berekend. Na het berekenen van een score voor alle mogelijke uitkomsten, kiest de computerspeler de zet die hem de hoogste score oplevert. De score van een spelbord wordt berekend als volgt:
 - Er staan 4 schijfjes van de computerspeler op een rij.
punten: +150
 - Er staan 3 schijfjes van de computerspeler op een rij (verticaal, horizontaal of diagonaal), en er is nog 1 lege plaats links, rechts of in het midden van deze rij.
punten: +10

- Er staan 2 schijfjes van de computerspeler op een rij (verticaal, horizontaal of diagonaal), en er zijn nog 2 lege plaatsen links, rechts of in het midden van deze rij.
punten: +1
- Er staan 3 schijfjes van de menselijke speler op een rij (verticaal, horizontaal of diagonaal), en er is nog 1 lege plaats links, rechts of in het midden van deze rij.
punten: -100
- Er staan 2 schijfjes van de menselijke speler op een rij (verticaal, horizontaal of diagonaal), en er zijn nog 2 lege plaatsen links, rechts of in het midden van deze rij.
punten: -10

Zorg ervoor dat elke zet van de computerspeler een geldige zet is. M.a.w er kan geen schijfje in een kolom gestoken worden die vol zit, een aambeeld kan enkel gespeeld worden als er nog aambeelden voor de computerspeler beschikbaar zijn enz.

1.3 Einde van het spel

Het spel eindigt wanneer het spelbord vol met schijfjes zit, of 1 van de spelers vier op een rij gehaald heeft. Elke keer dat een speler wint, zal deze speler een punt bijverdienen. Op het einde van het spel zal gevraagd worden of de speler nog een spel wilt spelen (de spelparameters besproken in 1.1 blijven hierbij hetzelfde).

1	2	3	4	5	6	7	
_	_	_	_	_	_	_	Computer: 1 []
_	_	_	_	_	_	_	Stijn: 3 [#]
_	_	_	O	X	_	_	
_	_	O	X	X	O	X	
_	_	X	O	O	O	X	
#	X	X	O	O	X	O	

Computer (X) koos kolom: 2

De computer wint!

Wil je nog een keer spelen?

2 Verloop van het spel

In deze sectie vind je een voorbeeld van een spelverloop. Hieruit kan je inspiratie halen voor de vragen die aan de speler worden gesteld etc.

1	2	3	4	5	6	7	
_	_	_	_	_	_	_	Computer: 1 [#, #]
_	_	_	_	_	_	_	Stijn: 3 [#,#]
_	_	_	_	_	_	_	
_	_	O	X	X	O	X	
_	_	X	O	O	O	X	
_	X	X	O	O	X	O	

Computer (X) koos kolom: 2
 wilt u een aambeeld gebruiken? N
 kies een kolom: 1

1	2	3	4	5	6	7	
_	_	_	_	_	_	_	Computer: 1 [#]
_	_	_	_	_	_	_	Stijn: 3 [#, #]
_	_	_	_	_	_	_	
_	_	O	X	X	_	X	
_	_	X	O	O	#	X	
O	X	X	O	O	X	O	

Computer (X) koos kolom: 6
 wilt u een aambeeld gebruiken? Y
 kies een kolom: 7

	1	2	3	4	5	6	7	
	_	_	_	_	_	_	_	Computer: 1 [#]
	_	_	_	_	_	_	_	Stijn: 3 [#]
	_	_	_	_	_	_	_	
	_	_	O	X	X	X	_	
	_	_	X	O	O	#	#	
	O	X	X	O	O	X	O	

Computer (X) koos kolom: 6
wilt u een aambeeld gebruiken?

3 Klassen

Uit de opgave kan je opmaken dat je bepaalde functionaliteit meermaals nodig hebt. Denk bijvoorbeeld aan het opvragen wat er zich op bepaalde coördinaten (rij x en kolom y) bevindt. Je hebt al minstens 2 klassen nodig (maar waarschijnlijk meer): Een klasse Spel die het gehele spel voorstelt, en een klasse SpelBord die de spelbord voorstelt. De klasse Spel kan dan bijvoorbeeld na elke beurt aan het SpelBord object vragen of er al een speler is met vier op een rij, of het spelbord al vol is enz.

Een SpelBord kan een 2 dimensionele array van Schijf-objecten bevatten, en onder andere methoden aanbieden om te kijken of op bepaalde coördinaten een Schijf-object staat, of om een Schijf-object op een bepaalde plaats op te halen. Een Schijf-object kan dan een gewoon schijfje van een speler zijn, of een aambeeld.

Nog enkele tips

- Lees de opgave heel aandachtig! Liefst meer dan eens, voor je nog maar begint met programmeren.
- Het is mogelijk om heel het programma in één main methode te maken, maar dit is een zeer slechte manier. Je hebt in de les geleerd over het ontwerpen van verschillende klassen, gebruik deze kennis om een duidelijk gestructureerde oplossing te programmeren.

- Je code zal (waarschijnlijk) niet van de eerste keer werken. Schrijf debug berichten naar het scherm, die zeggen wat er gebeurt in de code. Bijvoorbeeld, vooraleer een schijfje door de (moeilijke) computerspeler te laten zetten, schrijf je eerst de waarde van de score voor elke mogelijke zet naar het scherm. Indien deze fout zijn, weet je onmiddellijk dat er op dat punt een fout zit. Vergeet wel niet deze berichten te verwijderen als je indient. (Hoewel wat extra uitvoer véél minder erg is dan een oplossing die niet werkt!) Er zijn verschillende manieren om dit practicum op te lossen. Tijdens de mondelinge evaluatie zal je gevraagd worden waarom je bepaalde keuzes hebt gemaakt. Belangrijk hierbij is dat je kan verantwoorden waarom je een bepaalde keuze hebt gemaakt, wat de alternatieven zijn, en waarom jouw oplossing de beste is. Denk dus eerst goed na over je ontwerp, en mogelijke alternatieven!
- Bouw je programma stap voor stap op. Begin eerst met een SpelBord-klasse te maken, en het spelbord te tekenen. Zodra je dit goed getest hebt, laat je de speler een schijfje aan het spelbord toevoegen. Zodra dit getest is, kan je een aambeeld toevoegen, enz.
- Voor vragen kan je steeds terecht op het discussie forum op Toledo.