

FUNCTIONEEL ONTWERP_

Memory

M.C. van der Maas

21 augustus 2020

INHOUDSOPGAVE

1	OPDRACHTOMSCHRIJVING	4
2	GLOBALE FLOW	5
3	SCHERMONTWERPEN	6
	3.1 Startscreen	6
	3.2 Spelschermen	6
	3.3 Eindscreen	11
4	PRIORITERING	12

INLEIDING

Het vak Structured Programming – Application Development (SPAD) wordt afgesloten met een individuele opdracht en een bijbehorend assessment. Het is de bedoeling dat je dit functioneel ontwerp gebruikt als basis voor de realisatie van jouw opdracht.

Je vindt in dit document achtereenvolgens de opdrachtschrijving, de globale flow van het te realiseren programma, bijbehorende schermschetsen en een prioritering aan de hand van de MoSCoW methode.

Lees dit functioneel ontwerp goed door voordat je begint. De beschrijving en schermafbeeldingen zijn gebaseerd op de Must-haves en Should-haves. De Must-haves moeten minimaal worden gerealiseerd om een voldoende te kunnen halen.

Kijk goed welke producten je moet inleveren en welke eisen worden gesteld aan deze producten, inclusief jouw code. Deze informatie kun je vinden op Onderwijs Online en jouw docent licht dit toe. Vraag jouw docent om verheldering als je nog vragen hebt, iets onduidelijk vindt of iets niet kunt vinden.

Veel succes met jouw opdracht!

1 OPDRACHTOMSCHRIJVING

Iedereen kent wel het spel Memory. Dit is het spel waarbij een serie kaartjes met de blinde zijde naar boven liggen. Spelers draaien twee kaartjes om. Wanneer deze twee kaartjes hetzelfde beeld bevatten, heeft deze speler dit setje van twee kaarten gewonnen. In dit geval mag deze speler nog een keer. Wanneer de kaartjes verschillend zijn, worden ze weer omgedraaid en is de andere speler aan de beurt.

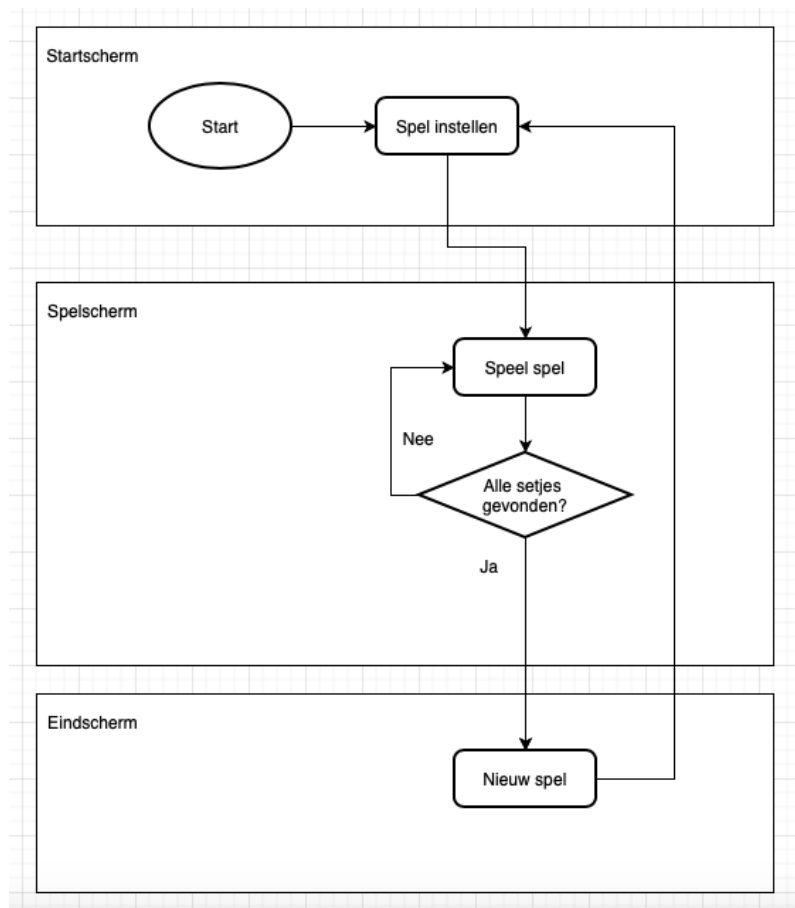
De casus is, om in Processing een memoryspel te realiseren dat als volgt functioneert:

Het spel vraagt met hoeveel setjes gespeeld moet worden. Bij elk setje hoort een plaatje dat dus bij twee kaarten gaat behoren. De kaarten worden in willekeurige volgorde verdeeld over het scherm en met de blinde kant naar boven toe getoond.

Het spel geeft Speler1 de beurt. Speler1 klikt op een kaart, waarna het spel het bijpassend plaatje toont. Speler1 selecteert een tweede kaart. Het spel toont ook dit plaatje. Het spel wacht twee seconden. Als beide plaatjes gelijk zijn, krijgt Speler 1 een punt (wordt getoond op scherm) en worden de beide kaarten verwijderd uit het spel. Als beide plaatjes ongelijk zijn, wordt de blinde kant van de kaartjes weer getoond en gaat de beurt naar speler2. Zo gaat het door tot alle kaartjes van het scherm verdwenen zijn. Dan wordt de uitslag getoond en aangegeven wie de winnaar is. Er verschijnt een keuze voor nog een spel of voor beëindigen.

2 GLOBALE FLOW

Globaal gezien zijn onderstaande situaties mogelijk. De pijlen geven aan welke situaties in elkaar over kunnen gaan. Deze situatieverandering is altijd een gevolg van een gebeurtenis (zoals een gebruiker die ergens op klikt).



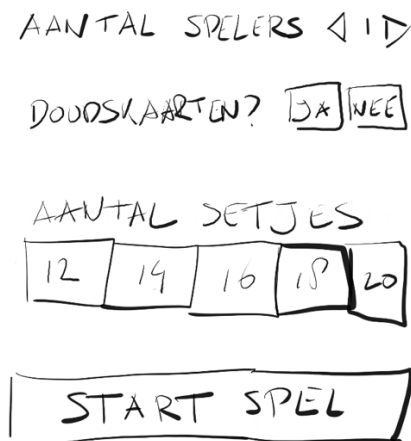
De bijbehorende schermontwerpen staan in het volgende hoofdstuk.

3 SCHERMONTWERPEN

Hieronder staan de schermontwerpen behorend bij verschillende momenten van het programma.

3.1 Startscherm

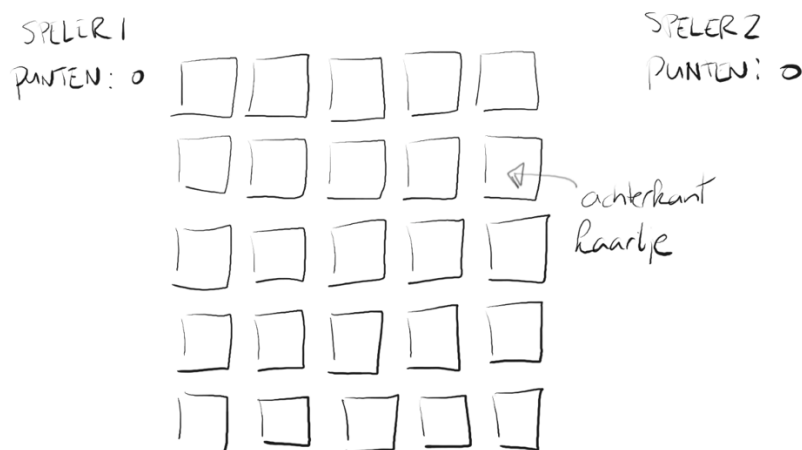
Zodra het programma gestart wordt zal er een startscherm te zien zijn, waarbinnen de speler onderstaande elementen van het spel kan instellen.



Figuur Startscherm

3.2 Spelschermen

Nadat er op de startknop binnen het startscherm wordt geklikt begint het spel met de door de speler gekozen instellingen. De speler krijgt dan het volgende scherm te zien.



Figuur Spelscherm

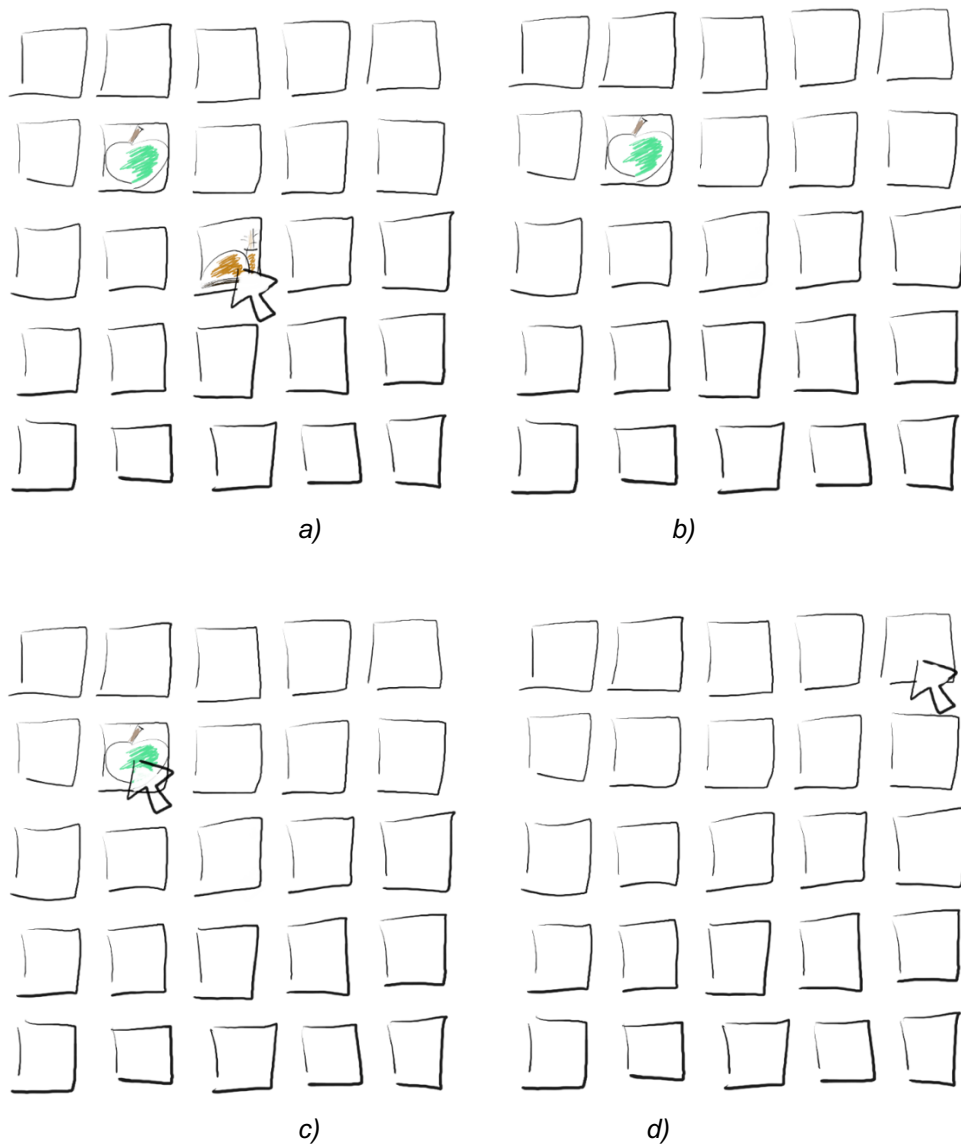
Bij aanvang van het spel zijn alle kaartjes omgedraaid. Alle achterkanten zijn gelijk zodat de speler de kaarten niet kan onderscheiden. De speler kan nu een setje zoeken door twee kaarten aan te klikken met de muis.



Figuur Per beurt 2 kaartjes kiezen

- a) *Muis is op eerste kaartje*
- b) *Op kaartje geklikt, kaartje omgedraaid*
- c) *Muis is op tweede kaartje*
- d) *Op kaartje geklikt, afbeeldingen zijn niet gelijk*

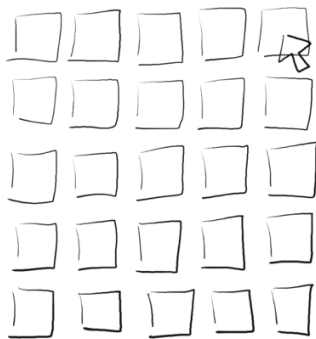
Wanneer de afbeeldingen van de omgedraaide kaartjes niet gelijk zijn, moeten de kaartjes weer worden omgedraaid.



Figuur Kaartjes terugdraaien

- a) Muis is op eerste kaartje
- b) Op kaartje geklikt, kaartje omgedraaid
- c) Muis is op tweede kaartje
- d) Op kaartje geklikt, kaartje omgedraaid

De speler verdient punten door een set van gelijke afbeeldingen te vinden.



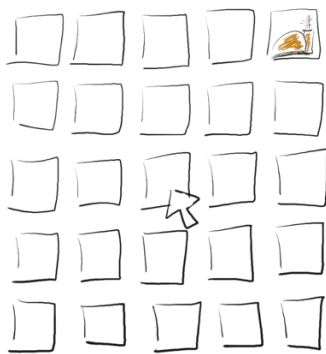
SPELER 2
PUNTEN: 0



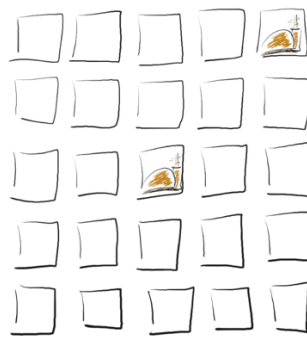
SPELER 2
PUNTEN: 0

a)

b)



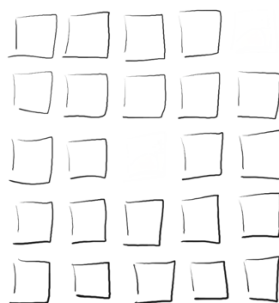
SPELER 2
PUNTEN: 0



SPELER 2
PUNTEN: 0

c)

d)



SPELER 2
PUNTEN: 1

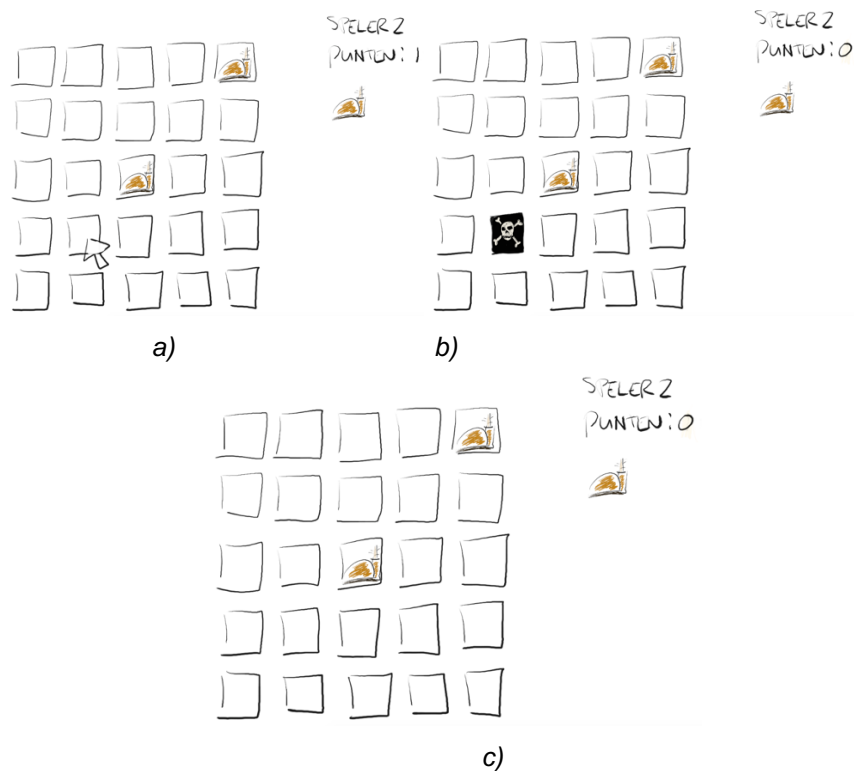


e)

Figuur Set gevonden

- a) Muis is op eerste kaartje
- b) Op kaartje geklikt, kaartje omgedraaid
- c) Muis is op tweede kaartje
- d) Op kaartje geklikt, kaartje omgedraaid
- e) Kaartjes zijn gelijk, set is uit het spel gehaald, speler heeft een punt

Kaarten worden altijd in een gelijkzijdig raster gelegd. In afbeelding van dit functioneel ontwerp is dat een raster van 5 x 5 kaarten. Een spel met 12 setjes bevat echter 24 kaarten. De overige kaarten zijn doodskaarten. In dit geval is dat dus slechts 1 kaart. De doodskaart staat op een willekeurige positie in het spel.



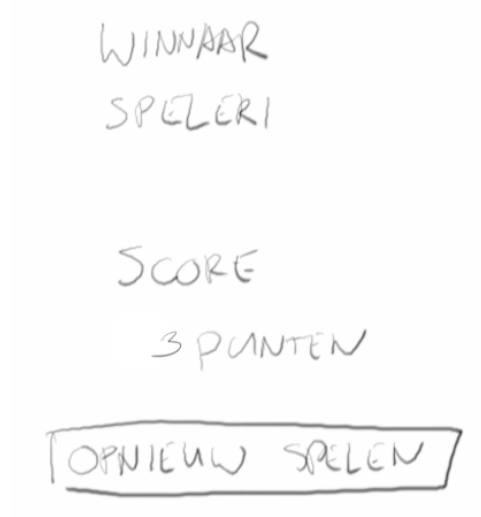
Figuur Doodskaat aanklikken

- a) De muis van speler 2 is op een kaartje
- b) Speler 2 klikt op de kaart, de kaart wordt omgedraaid en blijkt een doodskaat te zijn. Dit kost speler 2 een punt.
- c) Kaart is weer verborgen

Er wordt net zolang naar setjes gezocht totdat alle setjes zijn gevonden. Het spel is afgelopen.

3.3 Eindscherm

Nadat het spel is afgelopen wordt het eindscherm getoond. Het scherm toont de eindscore en een knop "opnieuw spelen". Bij een klik op deze knop komt de speler weer op het startscherm terecht.



Figuur Eindscherm

4 PRIORITERING

Hieronder staat de prioritering van de functionele eisen gegeven aan de hand van de MoSCoW methode.

M = Must have
S = Should have
C = Could have
W = Won't have

ID	Beschrijving	Prioriteit
FR01	Een 1 speler spel, waarbij de speler in zo min mogelijk beurten alle sets moet proberen te vinden.	M
FR02	Het spel wordt met de muis bestuurd.	M
FR03	Er wordt gespeeld met 12 setjes	M
FR04	Elk setje moet uniek zijn	M
FR05	De blinde zijde moet voor elke kaart gelijk zijn.	M
FR06	De kaarten zijn gelijkmatig verdeeld in een 5x5 speelveld en zijn willekeurig verdeeld.	M
FR07	Het spel bevat een doodskaart. Klikken op de doodskaart kost een punt.	M
FR08	De scores staan continu in beeld.	M
FR09	Er is een startscherm waarbinnen het aantal setjes in te stellen is: minimaal 12 en maximaal 32	S
FR10	Er is een eindscherm met de uiteindelijke uitslag en de mogelijkheid om opnieuw te beginnen.	S
FR11	De speler kan tegen een andere speler spelen.	S
FR12	Het spel bevat meerdere doodskaarten.	S
FR13	Het scherm toont de setjes die de speler gevonden heeft aan zijn kant van het speelveld.	S
FR14	De beurt van een speler gaat voorbij als de speler langer dan 10 seconden nadenkt.	S
FR15	Juist gekozen setjes bewegen met een animatie naar het gebied waar gevonden kaarten van de speler getoond worden.	S
FR16	Als een spel is gewonnen, dan gaat de speler door naar een volgend moeilijker level (meer setjes).	C
FR17	Er wordt een shuffle kaart toegevoegd. Wanneer op deze kaart wordt geklikt, worden alle kaarten die nog in het spel zitten opnieuw geschud.	C
FR18	De speler kan zijn naam ingeven op het startscherm. De ingegeven naam wordt getoond tijdens het spel.	C

FR19	Er wordt een verbruikte-beurten-lijst bijgehouden en opgeslagen in een savefile. Deze bevat de 10 beste spelers met het aantal verbruikte beurten en benodigde tijd tijdens hun beurten. Het eindscherm toont de lijst.	C
FR20	De speler kan kiezen voor een 2-player spel tegen een AI-bot die qua niveau in te stellen is.	C
FR21	Elke beurt wisselt de doodskaart met een andere kaart.	C
FR22	Alle elementen van het spel zijn schaalbaar t.o.v. het grafische scherm van processing.	C
FR23	Highscores worden bijgehouden en getoond in een apart scherm.	C

**OPEN UP
NEW HAN UNIVERSITY
OF APPLIED SCIENCES
HORIZONS.**