|  |  |
| --- | --- |
|  | Ontwerp subset |
|  |  |
| 1/25/2019 | Casus Gestructureerd Programmeren |
|  | Door: Rick Wubs (626010 Klas: ITA-1DA-D (SEB A DT) Docent: D. van Berkel |

Inhoudsopgave

[Opdrachtomschrijving 3](#_Toc4153314)

[Globale flow (tussen de schermen) 3](#_Toc4153315)

[Schermontwerpen 3](#_Toc4153316)

[Schermafbeelding 3](#_Toc4153317)

[Mogelijke acties 3](#_Toc4153318)

[Mogelijke gevolgen bij de acties 3](#_Toc4153319)

[Softwarebeschrijving 3](#_Toc4153320)

[Globale softwarestructuur 3](#_Toc4153321)

[Tabs 3](#_Toc4153322)

[Functies 3](#_Toc4153323)

[Programma flow (samenhang tussen methodes) 3](#_Toc4153324)

[Methoden en variabelen 3](#_Toc4153325)

[Namen, Parameters, Returnwaarden, Toelichting (tabel) 3](#_Toc4153326)

[Events 3](#_Toc4153327)

[Samenhang schermontwerp en variabelen 3](#_Toc4153328)

[Testplan 3](#_Toc4153329)

[Testcase 1 3](#_Toc4153330)

[Testcase 2 3](#_Toc4153331)

[Testcase 3 3](#_Toc4153332)

[Testcase 4 3](#_Toc4153333)

# Casusbeschrijving

De casus betreft het ontwerpen en implementeren van het spel subset, welke gebaseerd is op het spel Set.

Het spel bestaat uit kaarten, welke kunnen verschillen op een drietal eigenschappen, elk met een drietal mogelijke waarden:

1. De hoeveelheid figuren: 1, 2 of 3
2. De vorm van de figuren: driehoek, rechthoek of ellips
3. De kleur van de figuren: rood, groen of blauw

Elke combinatie komt slechts eenmaal voor. De volledige set aan permutaties van de genoemde eigenschappen levert een totale hoeveelheid van 27 unieke kaarten op.

Het spel toont in eerste instantie 9 kaarten, uitgelegd in een 3,3-matrix. Speler kan vervolgens 3 kaarten selecteren. Wanneer de drie kaarten geselecteerd zijn, wordt bepaald of deze selectie een Set is.

Een set houdt in dat de kaarten op elk van de drie eigenschappen ofwel verschillend, ofwel gelijk zijn. Ter illustratie: [1 rode driehoek, 2 blauw ellipsen, 3 groene rechthoeken] is een set, aangezien ze op alle eigenschappen (aantal, kleur en vorm) verschillen. [1 rode driehoek, 2 rode driehoeken, 3 rode driehoeken] zijn ook een set, aangezien ze op de eigenschappen kleur en vorm gelijk zijn, en op de eigenschap aantal verschillen. [1 rode driehoek, 1 groene driehoek, 1 blauw vierkant] is geen zet, omdat ze weliswaar gelijk zijn op aantal, verschillend op kleur, echter niet allemaal ofwel gelijk ofwel verschillend op vorm. Er zijn immers twee gelijk (driehoeken) en 1 verschillend (blauw).

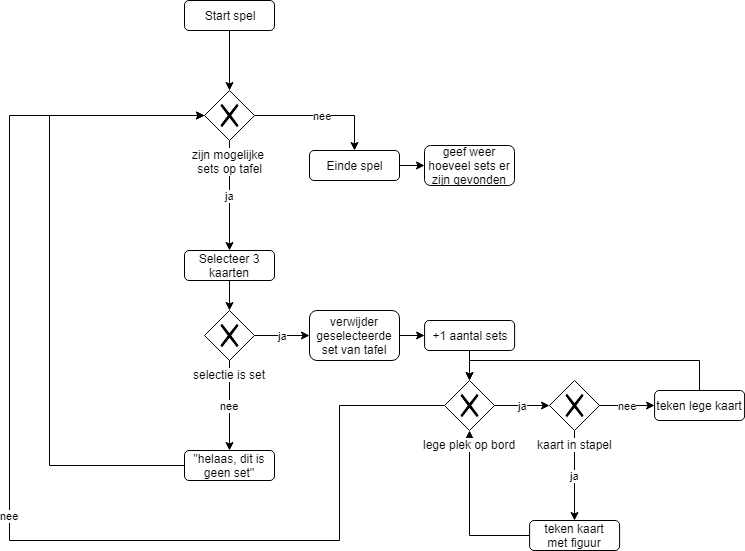
De minimale eisen zoals beschreven in de casus zijn als volgt:

1. Alle kaarten moeten aangemaakt worden.
2. De kaarten moeten geschud worden.
3. De kaarten moeten netjes op het beeld getekend worden.
4. De figuren worden netjes op de kaart getekend.
5. Een kaart mag maar eenmaal per spel gebruikt worden.
6. Er moet bijgehouden worden welke kaarten er in de stapel zitten.
7. Er moet bijgehouden worden welke kaarten op het scherm staan.
8. De kaarten moeten geselecteerd kunnen worden door de speler.
9. Bovengenoemde selectie moet zichtbaar zijn voor de speler.
10. Het moet steeds bekend zijn hoeveel mogelijke sets er op tafel liggen.
11. Wanneer er geen sets meer zijn is het spel afgelopen.
12. Er moet bijgehouden worden hoeveel sets de speler heeft gevonden.
13. Aan het einde van het spel moet aan de speler getoond worden hoeveel sets hij of zij heeft gevonden.

Geïmplementeerde extra functionaliteit zoals beschreven in optionele aanvullende eisen in de casus zijn als volgt:

1. Het eindresultaat van het spelletje wordt niet in de console, maar in het tekenvenster aan de speler gepresenteerd.

# Globale flow (tussen de schermen)



Bovenstaand diagram geeft een globale weergave van de flow van het programma weer. Niet alle functieaanroepen zijn hierin weergegeven; er is gekozen om enkel de voor de beslissingsboom belangrijke events weer te geven op een abstracter niveau.

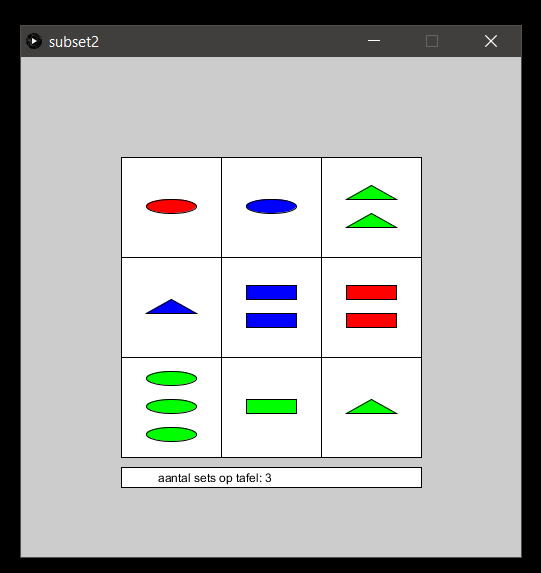
Wanneer het spel begint, er eerst gekeken wordt of er mogelijke sets op tafel zijn. Is dit niet het geval, dan is aan de “win”-conditie voldaan en eindigt het spel. Er wordt aan de speler weergegeven hoeveel sets hij of zij heeft behaald, in dit geval 0. Hiermee wordt tevens voldaan aan eis 13.

Als er mogelijke sets op tafel liggen, dan krijgt de speler kans om kaarten te selecteren. Wanneer er drie kaarten zijn geselecteerd, wordt gekeken of deze selectie een set betreft. Wanneer dit niet het geval is, wordt het bericht “helaas, dit is geen set” getoond. Wanneer dit wel het geval is, worden de geselecteerde sets van tafel verwijderd en wordt het aantal behaalde sets met 1 opgehoogt.

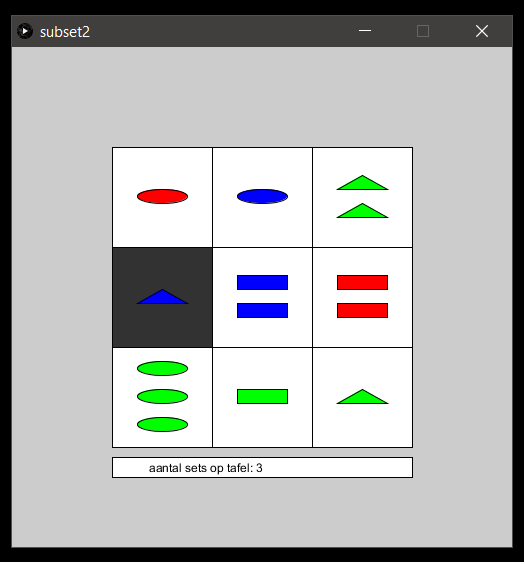
Vervolgens wordt gekeken of er een lege plek op het bord is. Zo niet, dan zijn er al genoeg kaarten gedeeld en gaan we terug naar de positie waarin gekeken wordt of er mogelijke sets op tafel liggen. Als er wel een lege plek op het bord is, wordt gekeken of er een kaart in de stapel met aangemaakte kaarten ligt. Zo ja, dan wordt er een kaart met figuur getekent. Zo niet, dan wordt er een lege kaart getekent. In beide gevallen wordt aansluitend wederom gekeken of er een lege plek op het bord is en zo nodig stappen herhaald.

# Schermontwerpen

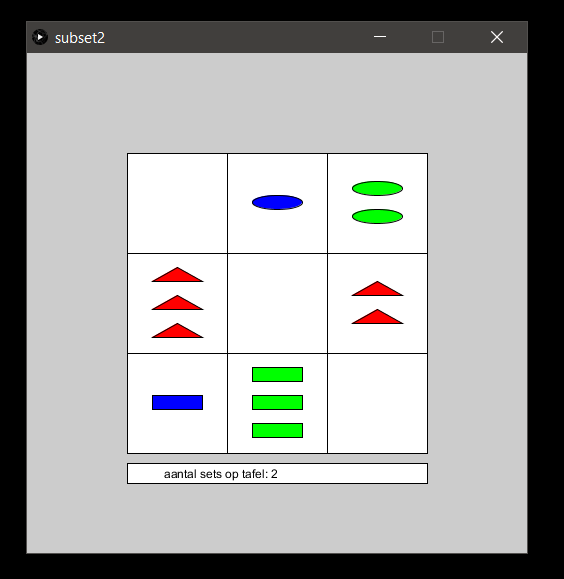
## Schermafbeeldingen



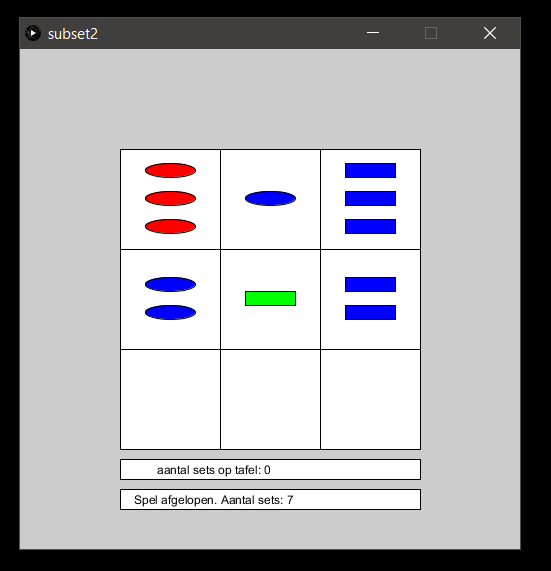
Scherm 1: geeft de startpositie weer. Er zijn 9 kaarten met figuren getekend, welke zijn opgehaald uit de array met alle aangemaakte kaarten. Tevens wordt weergegeven hoeveel mogelijke sets er op tafel liggen.



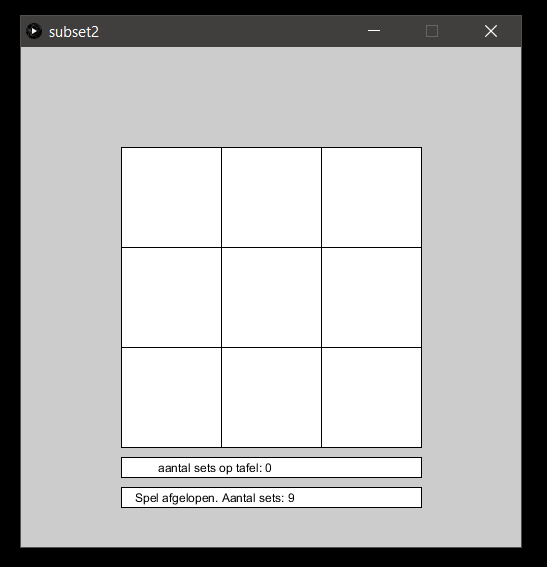
Scherm 2: wanneer de speler een kaart selecteert, wordt deze selectie voor hem of haar zichtbaar.



Scherm 3: wanneer de speler meerdere keren scherm 2 heeft gezien, komt er een moment waarop er nog sets op tafel liggen, maar geen kaarten meer in de array met aangemaakte kaarten zitten. Er worden dan lege kaarten getekend.



Scherm 4: Er zijn nog kaarten op het speelbord, maar er zijn geen kaarten meer in de stapel en geen sets meer op tafel. Het spel is afgelopen en het aantal behaalde sets wordt weergegeven.



Scherm 5: Er zitten geen kaarten meer in de array met aangemaakte kaarten en de speler heeft alle kaarten van tafel weg weten te spelen. Het spel is afgelopen en het aantal behaalde sets wordt weergegeven.

# Softwarebeschrijving

In deze sectie volgt een beschrijving van de technische aspecten van het ontworpen programma. De aangemaakte modules (tabs) met de functies worden eerst benoemd, vervolgens wordt de samenhang tussen de methoden beschreven. Vervolgens worden de methoden en variabelen benoemd en waar nodig toegelicht. Tevens worden de schermontwerpen nogmaals getoond, ditmaal in combinatie met de variabelen welke nodig zijn om deze items weer te geven.

Tot slot wordt er een korte reflectie gegeven over gemaakte keuzes en eventuele toekomstige verbeterpunten betreffende de structuur en implementatie van het ontwerp.

## Globale softwarestructuur

### Tabs

Het programma is opgedeeld in verschillende tabs. Elke tab betreft een afgebakend onderdeel binnen de structuur van het programma. Hieronder worden kort de tabs, alsook de functies per tab weergegeven. Meer informatie over de implementatie van de functies wordt in de sectie *methoden en variabelen* gegeven. De tabs zijn als volgt:

#### Subset2

Deze tab bevat globale variabelen, alsook algemene functionaliteit voor het spel. Het bevat de volgende functies:

1. Settings
2. Draw
3. mouseClicked

#### Kaarten

Deze tab bevat de methoden welke nodig zijn voor de functionaliteit omtrent kaarten. Deze tab bevat de volgende functies:

1. genereerKaarten
2. schudDeKaarten
3. vulBordMetKaarten
4. pakKaartVanStapel

#### Positie

Deze tab bevat de methoden welke nodig zijn voor de functionaliteit omtrent positie. Deze tab bevat de volgende functies:

1. voegSelectieToeAanGeselecteerdePosities
2. aantalKaartenInSelectieArray
3. eerstePositieLeeg
4. tweedePositieLeeg
5. derdePositieLeeg
6. maakGeselecteerdePositiesLeeg
7. berekenCoordinaten
8. aangekliktePositie
9. isMuisBinnenSpeelbord
10. rij
11. kolom

#### Sets

Deze tab bevat de methoden welke nodig zijn voor de functionaliteit omtrent sets. Deze tab bevat de volgende functies:

1. haalSetVanSpeelbord
2. krijgKaartenBijPosities
3. maakGeselecteerdeKaartenLeeg
4. nSetsOpTafel
5. isGeenLegeKaartOpTafel
6. isSet
7. drieKeerDezelfdeKaart
8. alleKaartenGelijk
9. alleKaartenVerschillend

#### Spelverloop

Deze tab bevat de methoden welke nodig zijn voor de functionaliteit omtrent het spelverloop. Deze tab bevat de volgende functies:

1. initializeerSpel
2. speelSpel
3. geenSetsMeerOpTafel
4. geenKaartenMeer
5. geenKaartenMeerInDeck
6. geenKaartenMeerOpTafel

#### tekenen

Deze tab bevat de functies welke nodig zijn voor de functionaliteit omtrent het tekenen van het speelbord, de kaarten en figuren. Deze tab bevat de volgende functies:

1. tekenBord
2. tekenEindeSpelTekst
3. tekenKaartOfLegePlek
4. tekenLegeKaart
5. tekenKaart
6. kaartGeselecteerd
7. tekenFiguren
8. tekenRechthoeken
9. tekenEllipsen
10. tekenDriehoeken
11. tekenRechthoek
12. tekenEllipse
13. tekenDriehoek
14. bepaalTypeFiguren
15. bepaalAantalFiguren
16. bepaalKleurFiguren
17. tekenSimpeleDriehoek

## Programma flow (samenhang tussen methodes)

Er is zoveel mogelijk getracht bij een methode alleen de voor die methode belangrijke begrippen te introduceren. Zo zijn er bijvoorbeeld functies welke alleen een simpele boolean testen en het resultaat teruggeven. Deze zijn onder andere geenSetsMeerOpTafel() en geenKaartenMeerInDeck(). Hiervoor is gekozen zodat functies als geenKaartenMeer() op de volgende manier kunnen functioneren:

boolean geenKaartenMeer() {

return geenKaartenMeerInDeck() && geenKaartenMeerOpTafel();

}

Voor deze functie zijn alleen deze twee condities van belang. *Hoe* bepaald wordt of er geen kaarten meer in de deck zitten, of *hoe* er bepaald wordt of er geen kaarten meer op tafel liggen, is voor deze functie totaal niet van belang. Mocht er op een bepaald moment andere condities zijn, bijvoorbeeld een aflegstapel, of meer kaarten tegelijk op tafel, dan kan deze functie zijn werk blijven doen en hoeft alleen de functionaliteit van de betreffende onderliggende functie aangepast te worden.

## Methoden en variabelen

### Globale variabelen

|  |  |  |
| --- | --- | --- |
| **Type** | **Naam** | **Omschrijving** |
| Static final int | ZWART | Waarde van de kleur zwart. Static final omdat deze waarde niet veranderd dient te worden. |
| Static final int | GRIJS | Waarde van de kleur grijs. Static final omdat deze waarde niet veranderd dient te worden. |
| Static final int | WIT | Waarde van de kleur wit. Static final omdat deze waarde niet veranderd dient te worden. |
| Static final int | ROOD | Waarde van de kleur rood. Static final omdat deze waarde niet veranderd dient te worden. |
| Static final int | GROEN | Waarde van de kleur groen. Static final omdat deze waarde niet veranderd dient te worden. |
| Static final int | BLAUW | Waarde van de kleur blauw. Static final omdat deze waarde niet veranderd dient te worden. |
| Int | schermBreedte | Breedte van het scherm. |
| Int | schermHoogte | Hoogte van het scherm. |
| Int | xPositieSpeelbord | X positie van het begin van het speelbord. Is schermBreedte / 5 omdat elke kaart ook deze afmeting heeft. 1 lege plek + 3 kaarten + 1 lege plek = 5 plekken totaal, elk met een breedte van schermBreedte / 5. |
| Int | yPositieSpeelbord | Y positie van het begin van het speelbord. Is schermHoogte / 5; zie xPositieSpeelbord. |
| Int | breedteSpeelbord | Breedte van het speelbord. Is 3 \* xPositieSpeelbord aangezien er aan beide kanten 1 lege plek moet zijn voor uitlijning. |
| Int | hoogteSpeelbord | Hoogte van het speelbord. Is 3 \* yPositieSpeelbord; zie breedteSpeelbord. |
| Int | xPositieKaart | De xPositie voor het tekenen van de kaart en berekenen van de positie. Is gelijk aan xPositieSpeelbord. |
| Int | yPositieKaart | De yPositie voor het tekenen van de kaart en berekenen van de positie. Zie xPositieKaart. |
| Int | breedteKaart | Breedte van de kaart. Zie xPositieSpeelbord. |
| Int | hoogteKaart | Hoogte van de kaart. Zie yPositieSpeelbord. |
| Int | breedteVorm | Breedte van de vorm. Is breedtekaart / 2 zodat deze goed uitgelijnd kan worden. |
| Int | hoogteVorm | Hoogte van de vorm. Is hoogtekaart / 7 zodat er 3 getekend kunnen worden met mooie uitlijning: lege plek + vorm1 + lege plek + vorm2 + lege plek + vorm3 + lege plek = 7 plekken. |
| String[] | eigenschappen | Geeft een array van 3 strings, elke string bestaande uit 3 characters. De eerste string is “123” en bevat het aantal te tekenen figuren. De tweede string is “rgb” en bevat de te tekenen kleur. De derde string is “RED” en bevat de te tekenen vorm. Deze is in hoofdletters om makkelijker te onderscheiden tussen “R”echthoek en “r”ood. |
| String[] | gedekteKaarten | Geeft een String array met 27 elementen, elke string bevat 3 characters, bijvoorbeeld “1rR” en geeft een kaart in de deck weer. |
| String[] | openKaarten | Geeft een string array met 9 elementen, elke string bevat 3 characters en geeft een getekende kaart weer. |
| String[] | geselecteerdeKaarten | Geeft een string array met 3 elementen, elke string bevat 3 characters en geeft een geselecteerde kaart weer. |
| Int | nGedekteKaarten | Geeft aan hoeveel kaarten er nog in de deck zitten. |
| Int[] | geselecteerdePosities | Geeft een array van integers, elke geeft een geselecteerde positie weer. |
| Int | nGeselecteerdePosities | Geeft aan hoeveel posities geselecteerd zijn. |
| Int | aantalSetsVanSpeler | Geeft aan hoeveel sets de speler op dit moment heeft. |

#### Functies

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tab** | **Naam** | **Parameters** | **Returns** | **Omschrijving** |
| Subset2 | Settings | - | Void | Initialisatie van het spel. Roept de functie initializeerSpel() aan. |
|  | Draw | - | Void | Functionaliteit van het spel. Loopt. Roept de functies tekenBord() en speelSpel() aan. |
|  | mouseClicked | - | Void | Functionaliteit voor als er met de muis wordt gedrukt. Voegt de selectie toe aan de geselecteerde posities en print output. |
| Kaarten | schudDeKaarten | - | String[] | Schud de kaarten door twee kaarten met elkaar te wisselen met een derde variabele als hulp. Doet dit 100 maal zodat het gehele array geschud is. Geeft gedekteKaarten terug |
|  | vulBordMetKaarten | - | Void | Loopt door de open kaarten. Als een van deze posities leeg is en er nog kaarten in de deck zitten, pakt hij een kaart van de stapel. Gebruikt de functie pakKaartVanStapel() |
|  | pakKaartVanStapel | - | String | Haalt de laatste kaart op uit de beschikbare kaarten uit de deck. Loopt door de open kaarten. Als er een lege plek is, voegt hij de kaart toe aan het array openKaarten. Tevens verlaagt hij het aantal kaarten in de deck met 1. |
| Positie | voegSelectieToe  aanGeselecteerde  Posities | - | Int[] | Beslist op basis van het aantal kaarten in het selectie array op welke positie in het array geselecteerde posities de selectie moet komen te staan. Wanneer er niet op een kaart wordt gedrukt geeft hij -1 terug, welke de waarde is van een leeg element in de array. Tevens print hij output “niet op een kaart geklikt”. Tot slot verhoogt hij het aantal geselecteerde posities en geeft dit array terug. |
|  | aantalKaartenIn  SelectieArray | - | Int | Geeft op basis van de methoden eerstePositieLeeg(), tweedePositieLeeg() en derdePositieLeeg() terug hoeveel kaarten er in het selectie array zitten. Als het array vol zit, geeft hij -1 terug. |
|  | eerstPositieLeeg | - | Boolean | Geeft weer of de eerste positie in het selectie array leeg is. |
|  | tweedePositieLeeg | - | Boolean | Geeft weer of de tweede positie in het selectie array leeg is. |
|  | derdePositieLeeg | - | Boolean | Geeft weer of de derde positie in het selectie array leeg is. |
|  | maakGeselecteerde  PositiesLeeg | - | Int[] | Maakt het selectie array leeg, door overal de waarde -1 in te zetten. |
|  | berekenCoordinaten | Bordpositie (int) | Int[] | Geeft op basis van de bordpositie de x en y coordinaat van deze positie terug. |
|  | aangekliktePositie | - | Int | Geeft de bordpositie terug als er binnen het speelbod is geklikt. Maakt hierbij gebruik van de functies isMuisBinnenSpeelbord(), rij() en kolom(). Als er buiten het speelbord is geklikt geeft hij -1 terug. |
|  | isMuisBinnenSpeelbord | - | Boolean | Geeft terug of de muis binnen het speelbord is of niet. |
|  | Rij | - | Int | Geeft het rijnummer terug op basis van mouseX |
|  | Kolom | - | Int | Geeft het kolomnummer terug op basis van mouseY |
| Sets | haalSetVanSpeelbord | - | Void | Zet de waarde van de geselecteerde open kaarten naar null. |
|  | krijgKaartenBijPosities | - | Void | Update de array geselecteerde kaarten op basis van de geselecteerde posities van open kaarten. |
|  | maakGeselecteerde  KaartenLeeg | - | Void | Maakt de array geselecteerde kaarten leeg door null waarden toe te kennen. |
|  | nSetsOpTafel | - | Int | Doorloopt met for-loops alle mogelijke combinaties van kaarten en bepaalt met behulp van de methoden isGeenLegeKaartOpTafel() en isSet() hoeveel sets er op tafel liggen. Geeft terug hoeveel sets er op tafel zijn. |
|  | isGeenLegeKaart  OpTafel | Kandidaatset  (String[]) | Boolean | Checkt of de kandidaatset geen lege kaart op tafel bevat. |
|  | isSet | Kandidaatset (String[]) | Boolean | Checkt eerst of er niet driemaal dezelfde kaart is gekozen met behulp van drieKeerDezelfdeKaart(). Kijkt vervolgens met behulp van alleKaartenGelijk() en alleKaartenVerschillend() of het een set betreft. Als dit voor een eigenschap waar is, wordt het aantal met 1 opgehoogd. Is het voor alle 3 eigenschappen waar, is het aantal gelijk aan 3 en geeft de functie “true” terug. |
|  | drieKeerDezelfdeKaart | Kandidaatset  (String[]) | Boolean | Geeft terug of er drie keer dezelfde kaart is gekozen. |
|  | alleKaartenGelijk | Kaart1 (char), kaart2 (char), kaart3 (char) | Boolean | Geeft terug of alle kaarten op een bepaalde eigenschap gelijk zijn aan elkaar. |
|  | alleKaartenVerschillend | Kaart1 (char), kaart2 (char), kaart3 (char) | Boolean | Geeft terug of alle kaarten op een bepaalde eigenschap verschillend zijn van elkaar. |
| Spelverloop | initializeerSpel | - | Void | Stelt de breedte van het scherm in. Genereert de kaarten, schud deze vervolgens, maakt de geselecteerde posities leeg en vult het bord met kaarten. |
|  | speelSpel | - | Void | als er geen kaarten meer zijn of geen sets meer op tafel zijn (met behulp van methoden met dezelfde namen) tekent het einde spel tekst. Zo niet, dan als het geselecteerde aantal posities 3 is, haalt hij de kaarten op bij de geselecteerde posities. Vervolgens kijkt hij of dit een set betreft. Zo niet, maakt hij de geselecteerde posities leeg, de array met geselecteerde kaarten en zet het aantal geselecteerde posities naar 0. Zo wel, dan haalt hij de set van het speelbord, maakt de geselecteerde posities leeg, maakt de geselecteerde kaarten leeg, zet het aantal geselecteerde posities naar 0, verhoogt het aantal sets van de speler en vult het bord met kaarten. |
|  | geenSetsMeerOpTafel | - | Boolean | Geeft weer of er nog sets op tafel zijn. |
|  | geenKaartenMeer | - | Boolean | Geeft weer of er geen kaarten meer in de deck en op tafel zijn |
|  | geenKaartenMeerInDeck | - | Boolean | Geeft weer of er nog kaarten in de deck zijn. |
|  | geenKaartenMeerOp  Tafel | - | Boolean | Checkt voor elke open kaart of deze een null-waarde bevat. Zo ja, wordt het aantal met 1 verminderd. Als alles leeg is, is het aantal 0. Als het aantal 0 is, geeft de functie “true” terug. |
| Tekenen | tekenBord | - | Void | Tekent het bord, en voor elke kaart tekent het een kaart of lege plek. Geeft tevens weer hoeveel sets er op dit moment op het bord liggen. |
|  | tekenEindeSpelTekst | - | Void | Tekent de tekst voor wanneer het spel eindigt. |
|  | tekenKaartOfLegePlek | Kaart (string), bordpositie (int) | Void | Tekent ofwel een lege plek op de betreffende bordpositie, ofwel een kaart. |
|  | tekenLegeKaart | Bordpositie (int) | Void | Berekent de coördinaten met behulp van betreffende functie, en tekent vervolgens een lege kaart op deze coördinaten. |
|  | tekenKaart | Kaart (string), bordpositie (int) | Void | Berekent de coördinaten met behulp van betreffende functie, en tekent vervolgens een kaart en met behulp van de betreffende functie tevens de juiste figuren. Wanneer de kaart geselecteerd is, wordt tevens de achtergrond grijs. |
|  | kaartGeselecteerd | Bordpositie(int) | Boolean | Geeft weer of een betreffende kaart geselecteerd is op basis van de betreffende array. |
|  | tekenFiguren | Kaart (string), x (int), y (int) | Void | Tekent de figuren op basis van de functie bepaalTypeFiguren(). |
|  | tekenRechthoeken | Kaart (string), x(int), y(int) | Void | Tekent rechthoeken op basis van het aantal figuren wat getekend moet worden. |
|  | tekenEllipsen | Kaart (string), x(int), y(int) | Void | Tekent ellipsen op basis van het aantal figuren wat getekend moet worden. |
|  | tekenDriehoeken | Kaart(string), x(int), y(int) | Void | Tekent driehoeken op basis van het aantal figuren wat getekend moet worden. |
|  | tekenRechthoek | Kaart(string), x(int), y(int) | Void | Tekent een rechthoek op basis van bepaalDeKleur() en de meegegeven parameters. |
|  | tekenEllipse | Kaart(string), x(int), y(int) | Void | Tekent een ellips op basis van bepaalDeKleur() en de meegegeven parameters. |
|  | tekenDriehoek | Kaart(string), x(int), y(int) | Void | Tekent een driehoek op basis van bepaalDeKleu() en de meegegeven parameters. |
|  | bepaalTypeFiguren | Kaart(string) | Char | Geeft aan welk type figuur getekend moet worden. |
|  | bepaalAantalFiguren | Kaart(string) | Char | Geeft aan welk aantal figuren getekend moet worden. |
|  | bepaalKleurFiguren | Kaart(string) | Int | Geeft aan welke waarde de kleur moet zijn wanneer een figuur getekend wordt. |
|  | tekenSimpeleDriehoek | X(int), y(int) | Void | Tekent een driehoek, in principe gelijk aan tekenDriehoek, echter zorgt deze functie ervoor dat slechts x en y meegegeven hoeven te worden. |

# Testplan

## Testcase 1: Schudden van de kaarten

**Beginsituatie:** de array met kaarten is aangemaakt **Eindsituatie:** de array met kaarten is geschud

|  |  |  |  |
| --- | --- | --- | --- |
| **Test** | **Actie** | **Verwachting** | **Resultaat** |
| 1.1 | Maak de array met kaarten tweemaal aan | We krijgen tweemaal dezelfde array terug | Komt overeen met verwachtingen |
| 1.2 | Maak de array met kaarten aan, schud eenmaal | We krijgen een array met dezelfde lengte terug, echter in andere volgorde | Komt overeen met verwachtingen |
| 1.3 | Maak de array met kaarten aan, schud meermaals | We krijgen de array met dezelfde lengte terug, echter telkens in andere volgorde | Komt overeen met verwachtingen |
| 1.4 | Schud de kaarten zonder de array met kaarten te populeren | Krijgen alleen null-waarden terug | Komt overeen met verwachtingen |

## Testcase 2: Vullen van het bord met kaarten

**Beginsituatie:** Er is een bord met 0 of meer kaarten **Eindsituatie:** Het bord bevat 1+ kaarten, of het spel eindigt.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test** | **Actie** | **Verwachting** | **Resultaat** |
| 2.1 | Bord met 9 kaarten, een selectie is geen set | Er worden geen kaarten toegevoegd of gewijzigd. | Komt overeen met de verwachtingen. |
| 2.2 | Bord met 9 kaarten, een selectie is een set, er zitten nog 3 kaarten in de stapel. | Er worden 3 kaarten getekend uit de array | Komt overeen met de verwachtingen. |
| 2.3 | Bord met 9 kaarten, een selectie is een set, er zitten N kaarten in de stapel, waar N < 3. | Er worden N kaarten toegevoegd uit de array, hierna worden er 9 – N lege kaarten getekend. | Komt overeen met de verwachtingen. |
| 2.4 | Bord met 9 kaarten, selectie is geen set, er zitten geen kaarten meer in de stapel | Er worden op de lege plekken alleen lege kaarten getekend. | Komt overeen met de verwachtingen. |
| 2.5 | Bord met 9 kaarten, selectie is een set, er zitten geen kaarten meer in de stapel | Er worden op de lege plekken alleen lege kaarten getekend. | Komt overeen met de verwachtingen. |
| 2.6 | Bord met < 9 kaarten, selectie is geen set, er zitten geen kaarten meer in de stapel | Er worden op de lege plekken alleen lege kaarten getekend. | Komt overeen met de verwachtingen. |
| 2.7 | Bord met <9 kaarten, selectie is een set, er zitten geen kaarten meer in de stapel | Er worden op de lege plekken alleen lege kaarten getekend. | Komt overeen met de verwachtingen. |
| 2.8 | Bord heeft 0 kaarten | Er worden geen kaarten meer getekend, het spel is voorbij | Komt overeen met de verwachtingen. |

## Testcase 3: Vullen van de array met mogelijke kaarten

**Beginsituatie:** array van lege strings  **Eindsituatie:** array van strings, gevuld met alle combinaties van kaarten

|  |  |  |  |
| --- | --- | --- | --- |
| **Test** | **Actie** | **Verwachting** | **Resultaat** |
| 3.1 | Print de array zonder deze te vullen | Geeft alleen null waarden weer | Komt overeen met de verwachtingen. |
| 3.2 | Vul de array, print deze vervolgens | We verkrijgen 27 kaarten, betreffende alle permutaties van 3 mogelijke eigenschappen | Komt overeen met de verwachtingen; elke combinatie komt slechts 1 keer voor en alle opties zijn gebruikt. |

# Conclusie

De combinatie van de rapportage, waarin zowel de structuur alsook de methoden en variabelen uitgebreid zijn beschreven en tevens testcases in zijn opgenomen, in combinatie met een werkend programma welke voldoet aan alle gestelde eisen, houdt in dat ik naar mijn mening aan de gestelde eisen als beschreven in de casus en het document *beoordeling beroepsproduct* heb voldaan.

Ik vond het een zeer leerzaam project. Ik heb nog niet veel programmeer-ervaring en het geeft veel voldoening een dergelijk zelfgeschreven programma tot stand te zien komen.

Er is tijdens het ontwikkelen van het programma getracht zoveel mogelijk de logica te verbergen waar nodig, zodat een functie alleen die taal gebruikt welke passend is bij het domein waarbinnen deze functie zich bevindt.

Echter door tijdgebrek is dit (nog) niet op alle plekken volledig doorgevoerd; er is gekozen eerst te zorgen voor een functioneel product en deze waar mogelijk te verbeteren. Ik merkte dat met name dit herschrijven van de code veel voldoening gaf. Wel hoop ik hier gedurende de studie meer *tools* voor te ontwikkelen; ik denk dat hier zeker nog verbetering mogelijk is.

Tevens stoor ik me mateloos aan het hebben van globale variabelen; ik verwacht dat met een op OOP gebaseerd programma ik veel van deze variabelen had kunnen verbergen voor functies welke deze niet nodig hebben. Echter ben ik nog niet overtuigd genoeg van mijn capaciteiten op het gebied van OOP om het programma volledig in dit paradigma te schrijven.

Tevens stoor ik mij aan een aantal switch-statements. Mijns inziens moeten hier veel betere oplossingen voor mogelijk zijn; echter doordat driehoeken op een andere manier getekend worden kon ik niet telkens dezelfde positie op de kaart gebruiken, waardoor het abstraheren van deze betreffende functies mij niet is gelukt.

Wel is mijn doel zodra ik wat tijd heb deze zaken nog aan te pakken; ik ben er van overtuigd dat ik hier nog winst kan behalen.