Hogeschool van Arnhem en Nijmegen

Ontwerp  
SubSet

Functioneel en technisch ontwerp van Subset

Alexander James Becoy

Studentnummer: 597880

3-11-2017

Versie 1.0

# 1. Inhoudsopgave

[**1. Inhoudsopgave**](#_yjkt7yclwinx) **1**

[**2. Opdrachtomschrijving**](#_9ngm6y7ydc4q) **2**

[**3. Requirements**](#_d39tmrkjhew) **2**

[3.1. Verplichte functionaliteit](#_oa36bmum90cw) 2

[3.2. Overige eisen](#_xmsf1dtko6q) 3

[3.3. Extra functionaliteit](#_nhnf8j4vkaak) 3

[**4. Globale Flow**](#_vx1y2plym95) **4**

[**5. Schermontwerpen**](#_cag7implcv05) **5**

[5.1. Start spel](#_iknu0l6sk1fg) 5

[5.2. Set maken](#_nmnrdpd0undi) 6

[5.2.1. Juiste set](#_v5nz39sxmx78) 7

[5.2.2. Set maken wanneer er geen gedekte kaarten meer zijn](#_gck6zvadyqcz) 8

[**6. Softwarebeschrijving**](#_o9kb0p7amk2s) **10**

[6.1. Globale softwarestructuur](#_bo90007wrkvr) 10

[6.2. Events, Methoden en Variabelen](#_y54i52kxxcn5) 11

[**7. Programma flow**](#_6ulppiyja18u) **13**

[**8. Testplan**](#_ullcounj39rz) **14**

[TEST CASE 1: Set maken](#_jrlb15ob9fgt) 14

[TEST CASE 2: Pauzeren](#_y0jecdl3ant) 15

[TEST CASE 3: Einde spel](#_2tu9twtcg6n3) 16

[TEST CASE 4: Schermwisselen](#_ej4zqlonlw2s) 17

[TEST CASE 5: Hulp vragen](#_6oisyyrqcewp) 19

[**9. Testrapportage**](#_ur3o7jwmul9y) **20**

[**10. Conclusie**](#_2qh879e069xj) **20**

# 2. Opdrachtomschrijving

De opdracht is het realiseren van een bordspel ‘Set’ naar een computerspel. Er zijn 81 unieke kaarten in het spel die op vier eigenschappen van elkaar kunnen verschillen:

* De hoeveelheid figuren: 1, 2 of 3.
* De vorm van de figuren: driehoek, rechthoek of ellips.
* De kleur van de figuren: rood, groen of blauw.
* En de vulling van de kleuren: leeg, gestreept of volledig.

34 verschillende eigenschappen = 81 unieke kaarten

Omdat elke kaart is uniek van andere kaarten, komt dus elke combinatie precies eenmaal voor. Bijvoorbeeld: een kaart met 1 groene lege driehoek, een met 2 blauwe gestreept ellipsen en een met 3 rode volledig rechthoeken.

Maar het spel dat gemaakt moet worden wordt in een makkelijker moeilijkheid gespeeld. In plaats van vier verschillende eigenschappen wordt er maar drie gespeeld. Dus zijn er maar 27 unieke kaarten.

33 verschillende eigenschappen = 27 unieke kaarten.

Er zijn steeds 9 kaarten tegelijk te zien op het scherm en het is de taak van de speler om daarin “sets” te vinden. Een set bestaat altijd uit drie kaarten. Drie kaarten vormen alleen een set, als voor elk van de drie eigenschappen afzonderlijk geldt de eigenschap ofwel op alle drie kaarten gelijk is, ofwel op elke kaart verschillend is.

# 3. Requirements

In de opdracht worden er requirements gegeven wat het spel moet voldoen.

## 3.1. Verplichte functionaliteit

Het spel moet in elk geval de volgende functionaliteit bevatten:

* *Alle kaarten moeten aangemaakt worden.*
* *De kaarten moeten geschud worden.*
* *De kaarten moeten netjes op het beeld getekend worden.*
* *De figuren op de kaarten worden netjes op de kaart getekend.*
* *Een kaart mag maar eenmaal in het spel gebruikt worden.*
* *De kaarten moeten geselecteerd kunnen worden door de speler en dat moet zichtbaar zijn voor de speler.*
* *Het moet steeds bekend zijn hoeveel mogelijke sets er op tafel liggen.*
* *Je moet bijhouden hoeveel sets de speler heeft gevonden en dit aan het einde de speler presenteren.*

## 3.2. Overige eisen

Verder zijn er non-functionele eisen dat het spel moet voldoen:

* Er is een ontwerp beschikbaar dat voldoet aan de in de course besproken voorwaarden.
* Je hebt je aan de course behandelde programmeerconventies gehouden, dus je hebt in elk geval zo veel mogelijk gebruik gemaakt van (herbruikbare) functies en variabelen en hebt je code goed gedocumenteerd.
* Je hebt je programma uitgebreid getest: zowel de losse functies als het eindproduct met interface.

## 3.3. Extra functionaliteit

In de toekomst kunt het spel uitgebreid worden met de volgende functionaliteit. Deze worden verdeeld op de belangrijkheid op basis van MoSCoW-methode.

|  |  |
| --- | --- |
| De mogelijkheid om een geselecteerde kaart te de-selecteren door erop te klikken. | **M** |
| Het eindresultaat van het spelletje wordt niet in het console, maar in het tekenvenster aan de speler gepresenteerd. De speler hoeft dan dus helemaal niet naar het console te kijken. | **S** |
| Real-time informatie in beeld over hoeveel kaarten er nog in de “stapel” zitten en hoeveel sets er in beeld te vinden zijn. | **S** |
| Een knop “Geef me een hint” waarmee twee kaarten uit een van de op het scherm aanwezige sets gemarkeerd worden. Gebruik van deze knop zou tot puntenaftrek kunnen leiden. | **S** |
| De mogelijkheid om met twee spelers om de beurt drie kaarten te selecteren en zo echt te kunnen winnen of verliezen. | **C** |
| Niet drie eigenschappen weergeven (aantal, vorm en kleur), maar vier zoals in het originele spel (verzin zelf de vierde eigenschap: het mag “vulling” zijn, maar iets anders mag ook). Het is in dat geval een goed idee om net zoals in het originele spel niet 9, maar 12 kaarten te tekenen. | **C** |
| Een knop “Ik zie geen set” waarmee drie extra kaarten op het scherm komen. Gebruik van deze knop zou tot puntenaftrek kunnen leiden. | **C** |

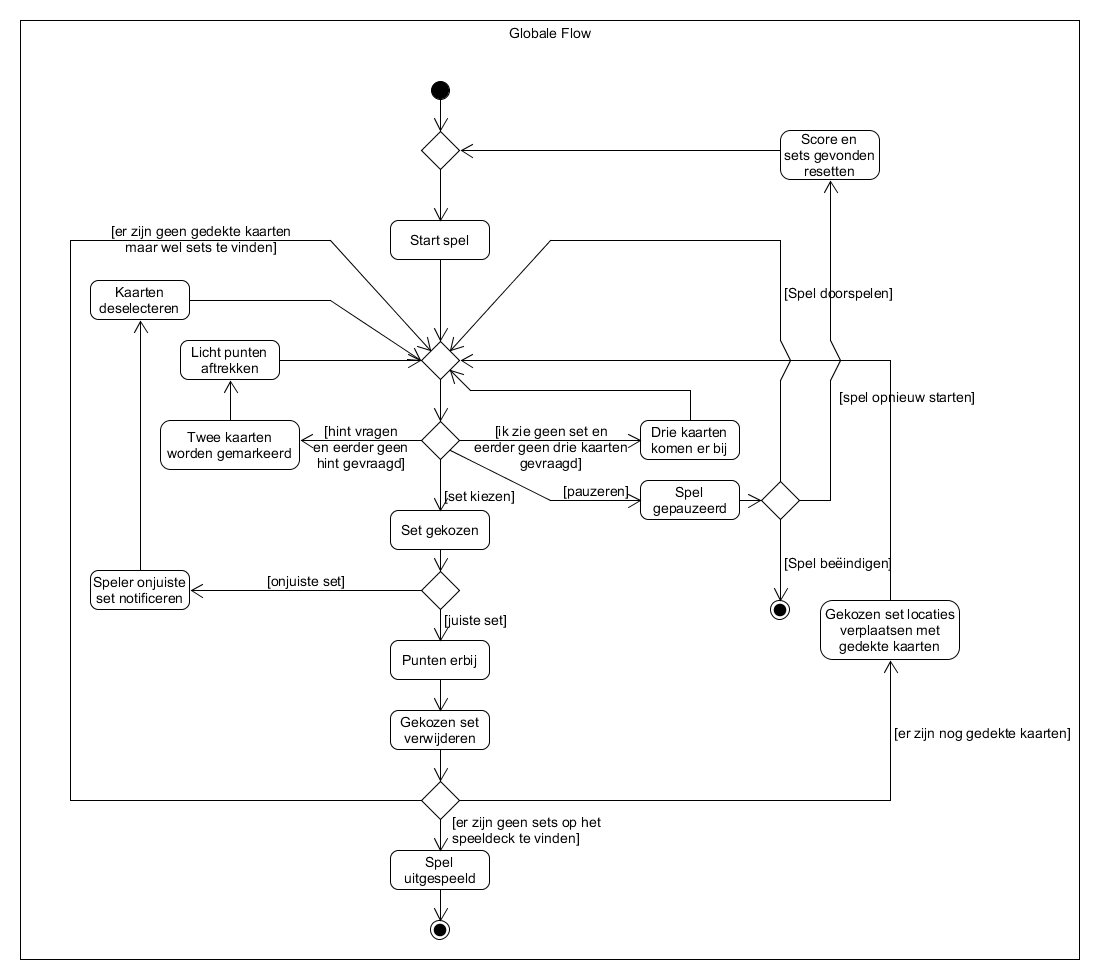
# 4. Globale Flow

In figuur 1 kunt u zien hoe een spel verloopt. De speler moet drie kaarten kiezen om een set te maken. Als de kaarten een onjuiste set vormen, wordt de speler genotificeerd dat het een onjuiste set is en worden de kaarten gedeselecteerd. Indien het wel een samen een juiste set vormen, worden er een aantal punten bij de score toegevoegd en worden de geselecteerde kaarten verwijderd.

Tijdens het spel heeft de speler ook de optie om te pauzeren, hints vragen en nog drie kaarten vragen. In de pauze-mode kan de speler kiezen om door te spelen, spel beëindigen en spel resetten. Als de speler om het spel opnieuw te spelen, worden ook de score en het aantal sets gevonden op nul gereset.

Als er nog gedekte kaarten zijn, dan vervangen het de plaatsen van de verwijderde kaarten over. Maar het spel blijft in iedergeval door.

Als er geen sets op het speeldeck te vinden zijn (ook al zijn er nog gedekte kaarten over zijn), is het spel uitgespeeld.



Figuur 1 - Globale Flow

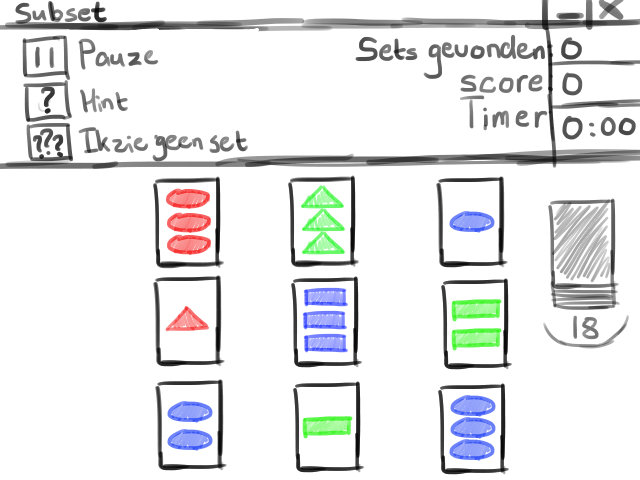
# 5. Schermontwerpen

Hier zijn schermontwerpen hoe het spel zult eruit zien. Elke mogelijkheid wordt vebeeld in een serie van ontwerpen; hoe het werkt en verloopt.

## 5.1. Start spel

Wanneer een spel is begonnen, start de timer te tikken. De speler bevindt de HUD(head-up display), 9 kaarten en een gestapelde gedekte kaarten. In de HUD zitten er hoeveel sets de speler heeft gevonden, aantal score de speler heeft gemaakt en 3 knoppen de speler kan gebruiken:

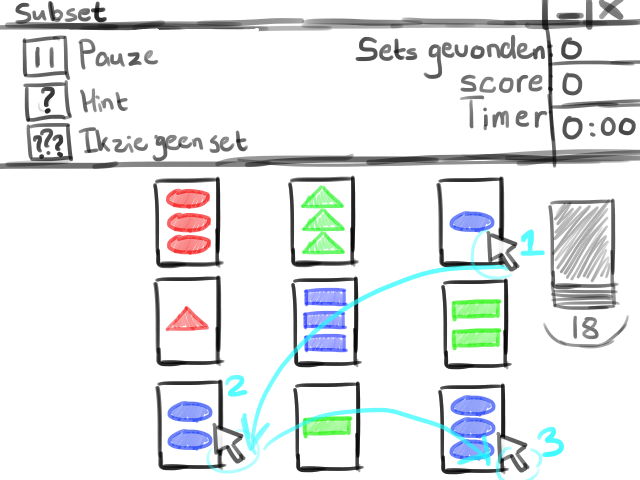
* Pauze: Het spel pauzeren of van daaruit het spel stoppen.
* Hint: Twee kaarten worden uit een van de op het scherm aanwezige sets gemarkeerd.
* Ik zie geen set: Drie extra kaarten komen op het scherm.

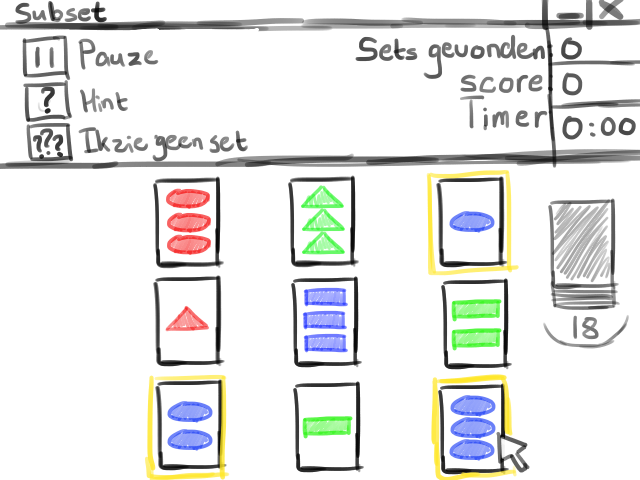


Figuur 2 - Spel begonnen

## 5.2. Set maken

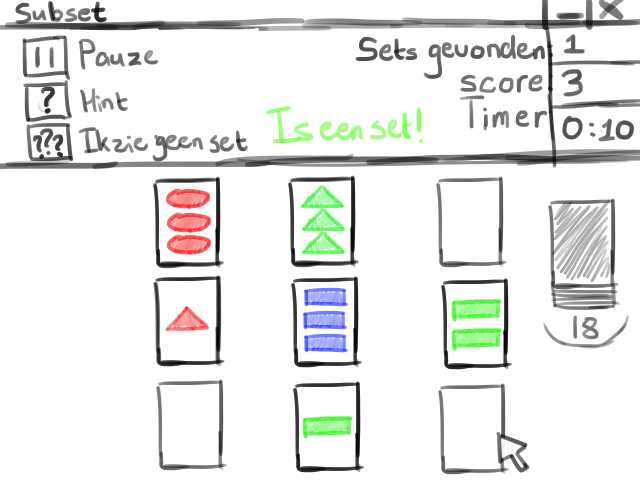
De speler klikt op 3 verschillende kaarten. De drie geselecteerde kaarten worden dan bekeken of de kaarten een juiste set maken.

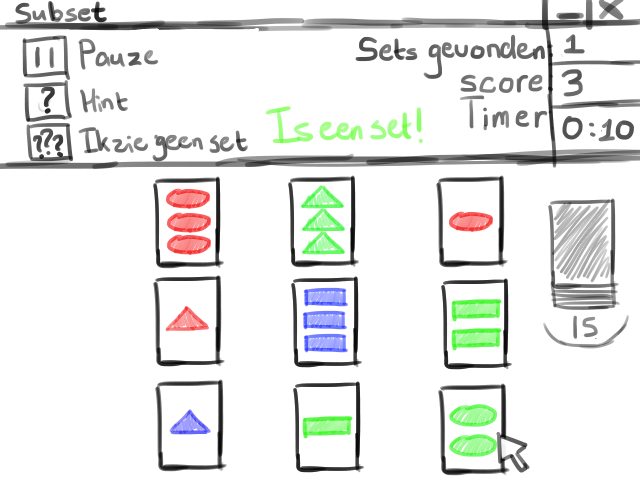




### 5.2.1. Juiste set

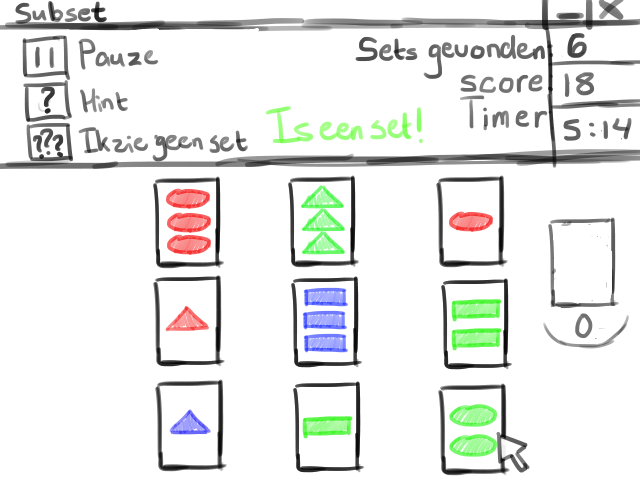
Als de drie geselecteerde kaarten een juiste set maken, worden dat kaarten verwijderd en verplaatsen de drie bovenste kaarten van de gedekte kaartenstapel de plaatsen over.



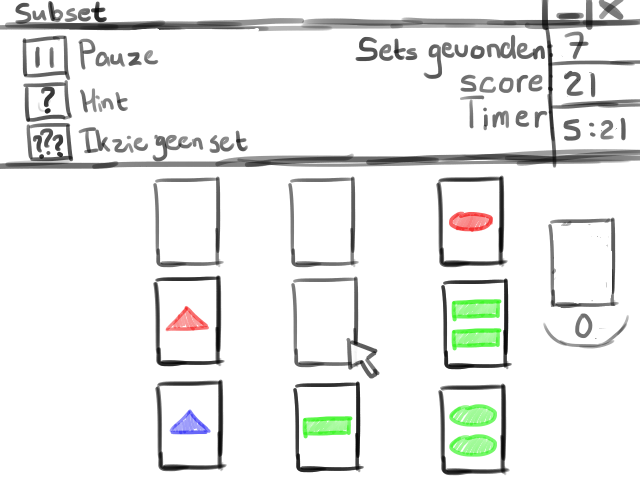
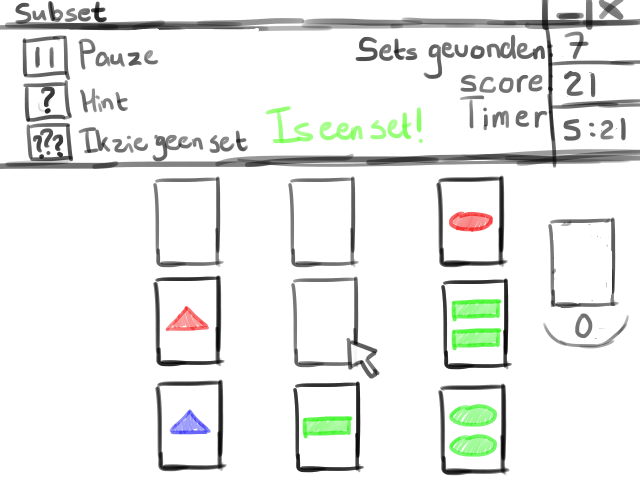
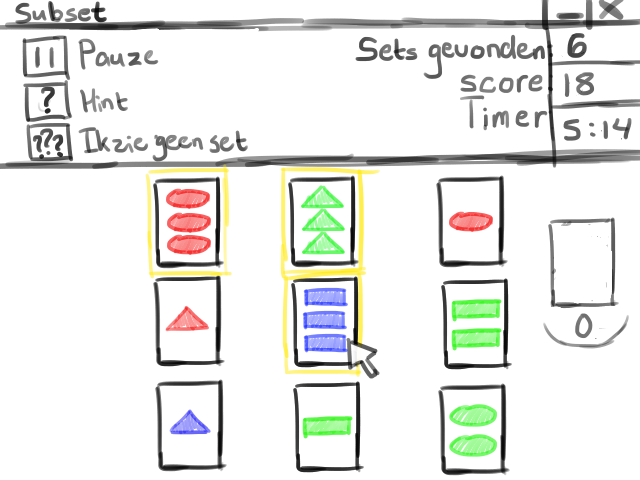


## 

### 5.2.2. Set maken wanneer er geen gedekte kaarten meer zijn







# 6. Softwarebeschrijving

## 6.1. Globale softwarestructuur

Het programma zal de volgende globale structuur bevatten:

* Module Instellingen: hierin wordt de scherminstellingen bijgehouden.
* Module Tekenen: hierin bevatten het tekenen van objecten en geheel.
* Module Events: hierin zullen de gebruikersacties worden afgevangen.
* Module Spelvarianten: hierin zitten elke methode hoe een spelvariant loopt.
* Module Algemeen: hierin zitten de algemene methoden en functies.
* Module Kaarten: hierzin zitten alle methoden met kaarten werken.

## 

## 

## 6.2. Events, Methoden en Variabelen

Hieronder kunt u een overzicht van allerlei events, methoden en variabelen die er nodig zijn om het spel te kunnen realiseren. In de toekomst zullen misschien de namen van de event/methode/variabele veranderen en nieuwe bijkomen.

|  |  |  |
| --- | --- | --- |
| **Module** | **Event/Methode/Variabele** | **Toelichting** |
| Instellingen | int schermBreedte | Variabele voor de breedte van het scherm. |
|  | int schermHoogte | Variabele voor de hoogte van het scherm. |
|  | final float MENUHOOGTE | Variabele voor de hoogte van de head-up display. |
|  | String[] eigenschappen =  {“123”, “rgb”, “red”} | Bevat vier strings die elk een eigenschap representeren. In elke string zitten de mogelijke waarden voor die eigenschap. |
| Kaarten | String[] genereerKaarten(String[] eigenschappen) | Bevat for-loops die over de eigenschappen itereren en op basis daarvan alle mogelijke kaarten samenstelt. |
|  | String pakKaartVanStapel() | Kopieert de achterste kaart uit gedekteKaarten (dat is dus degene met volgnummer nGedekteKaarten-1) naar een open plaats in openKaarten en verlaagt nGedekteKaarten met 1. |
|  | boolean isSet(String[] kandidaatset) | Geeft op basis van drie kaarten aan of het een set is. |
|  | int nSetsOpTafel() | Zet een teller op en kjkt voor elke combinatie van drie kaarten op tafel of het een set is (een for-loop voor de eerste kaart, met daarin een for-loop voor de tweede kaart, met daarin een for-loop voor de derde kaart). Zo ja, dan wordt de teller met 1 verhoogd. Het totaal wordt geretourneerd. |
|  | int aangekliktePositie() | Geeft het positienummer van de kaart die zich op positie (mouseX, mouseY) bevindt. Als zich daar geen kaart bevindt, kun je bijvoorbeeld -1 laten retourneren. |
|  | int[] coordinatenBijBordPositie(int bordpositie) | Geeft een array (lengte van 2) van integers terug met daarin de x- en y-coordinaat behorende bij de bordpositie. Je gebruikt uiteraard width en height om deze positie te helpen bepalen. |
|  | String[] kaartenSchudden(String[] kaarten) | Husselt de kaartenstapel in willekeurig volgorde. |
| Tekenen | void tekenBord() | Het tekenen van het hele bord wanneer er een verandering of meer worden gemaakt. Bijvoorbeeld; er worden nieuwe kaarten getoond, er wordt een kaart geselecteerd, etc. |
|  | void tekenMenu() | Het tekenen van de head-up display boven het bord. Daar wordt de drie knoppen, de timer en score getoond. |
|  | void tekenKaart(String kaart, int bordpositie) | Tekent de kaart op basis van de stringrepresentatie en de positie op het bord. |
| Spelvarianten | int score | Houdt het aantal score bij. |
|  | int setsGevonden | Houdt het aantal sets die gevonden zijn bij. |
|  | String[] gedekteKaarten | Bevat alle kaarten die nog niet in het spel zijn. Is dus 27 elementen lang, of in de originele spelvariant 81 elementen lang. |
|  | int nGedekteKaarten | Geeft aan hoeveel gedekte kaarten er nog zijn. Zodra er kaarten van de “stapel” naar het scherm gaan, neemt dit aantal dus af. |
|  | String[] openKaarten | Bevat de kaarten die op het scherm staan. |
|  | int[] geselecteerdePosities | Hierin wordt bijgehouden welke van de kaarten in openKaarten is/zijn geselecteerd door de speler. Kan drie elementen bevatten. |
|  | int nGeselecteerdePosities | Geeft aan hoeveel kaarten er geselecteerd zijn. |
| Algemeen | boolean muisBovenRect  (float x, float y, float breedte, float hoogte) / boolean muisBovenCirkel  (float x, float y, float breedte, float hoogte) | Bekijkt of de muis boven een figuur hovert. |

# 

# 7. Programma flow

Hieronder staat een pseudocode hoe alle methoden elkaar samenhangen.

setting()

size(schermBreedte, schermHoogte);

draw()

tekenBord();

beginSpel()

gedekteKaarten = kaartenSchudden(genereerKaarten(eigenschappen));

openKaarten = pakKaartVanStapel();

tekenBord()

tekenMenu();

tekenKaarten();

if(nGeselecteerdePosities > 3)

if(isSet(geselecteerdePosities))

score++;

setsGevonden++;

openKaarten = pakKaartVanStapel();

tekenMenu()

rect(0, 0, schermBreedte, MENUHOOGTE);

text(“Sets op tafel” + nSetsOpTafel);

MouseClicked()

geselecteerdePosities = coordinatenBijBordPositie(aangekliktePositie());

# 

# 

# 8. Testplan

Onder staan er testcases die ontworpen zijn om de functionaliteit van het spel te kunnen testen. Elke test wordt uitgevoerd door een andere, zodat de resultaten niet eenzijdig zijn.

## TEST CASE 1: Set maken

|  |  |
| --- | --- |
| FUNCTIE ONTWIKKELD DOOR | Alexander James Becoy |
| TEST UITGEVOERD DOOR | Robert van Eijk |
| DATUM | 3/11/2017 |
| BEGINSITUATIE | Spel is opgestart |
| EINDSITUATIE | Een juiste set is aangemaakt, dus punten en het aantal sets gevonden worden bijgeteld. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TEST | VERIABELE OF  SELECTIE | WAARDE | VERWACHT  RESULTAAT | WERKELIJK  RESULTAAT |
| 1.1 | isSet() | Een juiste combinatie van drie kaarten worden geselecteerd. | De kaarten worden verwijderd. | Kaarten worden verwijderd. |
| 1.2 | isSet() | Een juiste combinatie van drie kaarten worden geselecteerd. | De kaarten worden verwijderd en drie kaarten van het stapel nemen het plaats over. | Kaarten worden verwijderd en nieuwe kaarten komen erbij. |
| 1.3 | isSet() | Een onjuiste combinatie van drie kaarten worden geselecteerd. | De kaarten worden gedeselecteerd. | Mijn gekozen kaarten worden deselecteer en er wordt getoond dat het geen set is. |
|  |  |  | De speler wordt genotificeerd dat het geen set is. |  |

## TEST CASE 2: Pauzeren

|  |  |
| --- | --- |
| FUNCTIE ONTWIKKELD DOOR | Alexander James Becoy |
| TEST UITGEVOERD DOOR | Dirk Lukassen |
| DATUM | 3/11/2017 |
| BEGINSITUATIE | Spel aan het spelen |
| EINDSITUATIE | Spel wordt gepauzeerd en kan de speler het spel verder spelen, opnieuw beginnen of stoppen. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TEST | VERIABELE OF  SELECTIE | WAARDE | VERWACHT  RESULTAAT | WERKELIJK  RESULTAAT |
| 2.1 | hoverPauzeKnop() | De speler drukt op de pauzeerknop. | Het spel wordt gepauzeerd en de kaarten en menuknoppen zijn niet te drukken. | Het spel wordt gepauzeerd en de kaarten en menuknoppen zijn niet te drukken. |
| 2.2 | hoverVerderKnop() | De speler drukt op de “Verder spelen” knop. | Het spel gaat door. | Ik kan doorgaan met spelen. |
| 2.3 | hoverOpnieuwKnop() | De speler drukt op de “Opnieuw spelen” knop. | Het spel begint opnieuw. Punten en aantal sets gevonden worden terug op 0 gezet. | het spel wordt gereset. |
| 2.4 | hoverStoppenKnop() | De speler drukt op de “Stoppen” | Het spel wordt gestopt en de speler wordt gebracht naar het hoofdmenu. | Ik word gebracht naar het hoofdmenu. |

## 

## 

## TEST CASE 3: Einde spel

|  |  |
| --- | --- |
| FUNCTIE ONTWIKKELD DOOR | Alexander James Becoy |
| TEST UITGEVOERD DOOR | Ronald Baauw |
| DATUM | 3/11/2017 |
| BEGINSITUATIE | Er zijn geen sets meer. |
| EINDSITUATIE | Speler krijgt de resultaten en kan naar het hoofdmenu gebracht worden. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TEST | VERIABELE OF  SELECTIE | WAARDE | VERWACHT  RESULTAAT | WERKELIJK  RESULTAAT |
| 3.1 | spelGewonnen() | Er zijn geen kaarten in de stapel meer. | Speler wordt genotificeerd met de status “Game Over” | Ik word genotificeerd met Game Over. |
| 3.2 | spelGewonnen() | Er zijn nog kaarten in de stapel | Speler wordt genotificeerd met de status  “Er zijn geen sets meer” | Ik word genotificeerd met “Er zijn geen sets meer”. |
| 3.3 | spelGewonnen() | Er zijn geen kaarten in de stapel meer en de speler drukt op “Terug naar het hoofdmenu”. | Speler wordt genotificeerd met de status “Game Over” en wordt naar het hoofdmenu gebracht. | Ik word genotificeerd met Game Over en gebracht naar het hoofdmenu. |
| 3.4 | spelGewonnen() | Er zijn nog kaarten in de stapel | Speler wordt genotificeerd met de status  “Er zijn geen sets meer” en wordt naar het hoofdmenu gebracht. | Ik word genotificeerd met “Er zijn geen sets meer” en gebracht naar het hoofdmenu. |

## 

## TEST CASE 4: Schermwisselen

|  |  |
| --- | --- |
| FUNCTIE ONTWIKKELD DOOR | Alexander James Becoy |
| TEST UITGEVOERD DOOR | Julian de Bruin |
| DATUM | 3/11/2017 |
| BEGINSITUATIE | De applicatie is opgestart. |
| EINDSITUATIE | De speler kan door verschillende schermen wisselen. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TEST | VERIABELE OF  SELECTIE | WAARDE | VERWACHT  RESULTAAT | WERKELIJK  RESULTAAT |
| 4.1 | int scherm | Speler drukt op de knop “Spelen” | De speler kan vanuit drie spelvarianten kiezen. | Ik kan van drie spelvarianten kiezen. |
| 4.2 | int scherm | Speler drukt op de knop “Verhaal” | De speler kan de verhaal achter het spel lezen. | Ik kan achterkomen hoe het spel is uitgevonden. |
| 4.3 | int scherm | Speler drukt op de knop “Regel” | De speler kan het regel hoe het spel werkt lezen. | Ik kan weten hoe het spel ongeveer werkt. |
| 4.4 | int scherm | Speler drukt op de knop “Één-speler SubSet” | De speler begint met SubSet spelen. | Ik kan in mijn eentje SubSet spelen. |
| 4.5 | int scherm | Speler drukt op de knop “Twee-speler SubSet” | De speler begint met SubSet spelen. | Ik kan samen tegen een andere SubSet spelen. |
| 4.6 | int scherm | Speler drukt op de knop “Één-speler Set” | De speler begint met Set spelen. | Ik kan in mijn eentje de originele Set spelen. |
| 4.7 | int scherm | Speler drukt op een terugknop. | De speler wordt naar het vorige scherm gebracht. | Ik word gebracht naar het vorige scherm. |
| 4.8 | int scherm | Speler drukt op een stopknop. | De applicatie wordt gestopt. | Het spel is afgesloten. |

## TEST CASE 5: Hulp vragen

|  |  |
| --- | --- |
| FUNCTIE ONTWIKKELD DOOR | Alexander James Becoy |
| TEST UITGEVOERD DOOR | Martijn Zwaan |
| DATUM | 3/11/2017 |
| BEGINSITUATIE | De speler is bezig met het spel spelen. |
| EINDSITUATIE | De speler kan hints vragen en kaarten van het stapel pakken en sets maken. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TEST | VERIABELE OF  SELECTIE | WAARDE | VERWACHT  RESULTAAT | WERKELIJK  RESULTAAT |
| 2.1 | boolean hintsGegeven | De speler drukt op de hintknop. | Twee kaarten vanuit de open kaarten worden geselecteerd en 5 punten worden afgetrokken. | Ik kan 2 kaarten zien die een mogelijke set kan maken. |
| 2.2 | boolean hintsGegeven | De speler drukt op de hintknop, maar een hint is al gegeven. | Er wordt niks veranderd. | Dezelfde 2 kaarten blijft geselecteerd. |
| 2.3 | openKaarten[] | De speler drukt op de kaarten stapel. | Drie nieuwe kaarten worden bijgevoegd en 3 punten worden afgetrokken. | Er komen 3 kaarten vanuit het stapel op het speeldeck. |
| 2.4 | openKaarten[] | De speler drukt op de kaarten stapel, maar er zijn al drie kaarten toegevoegd. | Er wordt niks veranderd. | Er blijven 12 ( of bij Set 15) kaarten op het scherm. |

# 9. Testrapportage

De werkelijke resultaten vanuit mijn testcases bleken bijna hetzelfde hoe ik het heb verwacht. Dus hoeft er niets te veranderen.

# 10. Conclusie

De kaarten kunnen worden geselecteerd en gedeselecteerd om een set te maken. Het systeem bekijkt of het een juiste set is. In het begin van het spel worden de kaarten geschud totdat er minimaal 1 mogelijke set op het scherm zitten. Tijdens het spel kan de speler zien hoeveel mogelijke sets er op het scherm zitten. Ook kan de het spel pauzeren en daarna verder spelen, opnieuw spelen of stoppen, hints vragen en 3 nieuwe kaarten pakken.

Na het spelen van het spel krijgt de speler de resultaten hoeveel score hij/zij heeft gehaald en het aantal gevonden sets.

Spelers kunnen ook tegen elkaar spelen en wie het meeste punten scoort wint, daarna krijgen zij allebei de resultaten te zien hoe goed zij hebben gedaan. Ook kunt de speler de originele Set spelen, precies hoe zoals SubSet werkt maar met 1 extra eigenschap.

Door het analyseren en ontwerpen van het spel geven mij het een betere beeld hoe het spel werkt en hoe ik het wilt realiseren. Zonder dat moeite, geeft het mij nog veel meer moeite en energie om te weten hoe ik het wil maken.