



BEROEPSPRODUCT

SubSet

Erik Knaake

598368

I1E

SPD

Opdrachtgever: Leo van der Berge

Versie: 1.4

1. Inhoud

1.	Inhoud	2
2.	Inleiding	3
3.	Functioneel ontwerp	4
3.1.	Spelregels subset	4
3.2.	Requirements	5
3.3.	Globale flow	6
3.4.	Flow van het programma	7
3.4.1.	Start van het spel	7
3.4.2.	Verloop van een spel	8
3.4.3.	Einde spel	9
3.5.	Schermontwerpen	10
3.5.1.	Spel	10
4.	Technisch ontwerp	12
4.1.	Lijst van variabele, functies en events	12
4.2.	Lijst van te maken globale variabelen:	14
4.3.	Lijst van te maken functies:	15
4.4.	Activity diagrammen	21
4.4.1.	schudKaarten	21
4.4.2.	isSet	22
4.4.3.	aantalSetsOpTafel	23
4.4.4.	tekenKaart	24
4.4.5.	tekenBord	25
4.4.6.	tekenFooter	25
4.4.7.	verwerkKlik	26
4.4.8.	vervangSetMetNieuweKaarten	27
5.	Test plan	28
6.	Conclusie	31
7.	Figuren	32
8.	Bibliografie	32

2. Inleiding

Na de vraag van de opdrachtgever om het spel subset te programmeren, is eerst dit rapport geschreven. Subset is een spel dat erg veel op set lijkt. Het is de bedoeling dat een speler sets vind. Dit rapport is in een aantal onderdelen opgedeeld, het functioneel ontwerp, het technisch ontwerp en het test plan. Wat subset is zal verder worden uitgelegd in het functioneel ontwerp.

Het functioneel ontwerp zal inzicht geven in de opdracht op een manier dat het voor de opdrachtgever duidelijk is dat de opdracht juist is begrepen. Het technisch ontwerp zal ervoor zorgen dat een programmeur de opdracht juist kan uitvoeren. Het test plan zal ervoor zorgen dat de geschreven software naar behoren zal werken.

3. Functioneel ontwerp

In dit hoofdstuk wordt besproken wat er wordt gemaakt. Eerst wordt er besproken welke regels het spel subset kent. Vervolgens wordt er besproken welke eisen er zijn gesteld door de opdrachtgever. Tot slot wordt er besproken hoe de schermen er uit komen te zien en hoe deze schermen interacteren.

3.1. Spelregels subset

Bij het spel subset zijn er 27 kaarten in het spel. Elke kaart heeft 3 attributen, een kleur, een vorm en een aantal symbolen. In het begin van het spel worden deze kaarten geschud. Hierna worden er 9 kaarten op de tafel gelegd. Het is de bedoeling dat een speler hierin een set vindt. Een set bestaat uit 3 kaarten waarvan elke eigenschap óf over elke kaart overeenkomt, óf voor elke kaart verschillend is. Als de speler een set heeft gevonden worden er 3 nieuwe kaarten op de tafel gelegd. Als er geen kaarten meer op de stapel liggen wordt dit niet gedaan. Het spel is afgelopen als er geen sets meer op de tafel liggen.

Uit deze spelregels en de casus beschrijving (HAN, 2017) van subset is er een lijst van requirements opgesteld.

3.2. Requirements

In dit subhoofdstuk volgt een omschrijving van de opdracht in de vorm van de gestelde eisen. Dit is gedaan met behulp van MoSCoW, een manier van prioriteren. De M staat voor must, de S voor should, de C voor could en de W voor would. De M heeft dus de hoogste prioriteit en de W de laagste. De M'en zijn te zien als functionaliteiten die per se nodig zijn om het spel te kunnen spelen. In de praktijk zijn alle requirements tot de woulds verplicht, er is gekozen om de shoulds en coulds wel in te delen, omdat dit meteen tot een stappenplan leidt. Verder zijn de eisen genummerd zodat later nog makkelijker terug te vinden is over welke eis we het hebben.

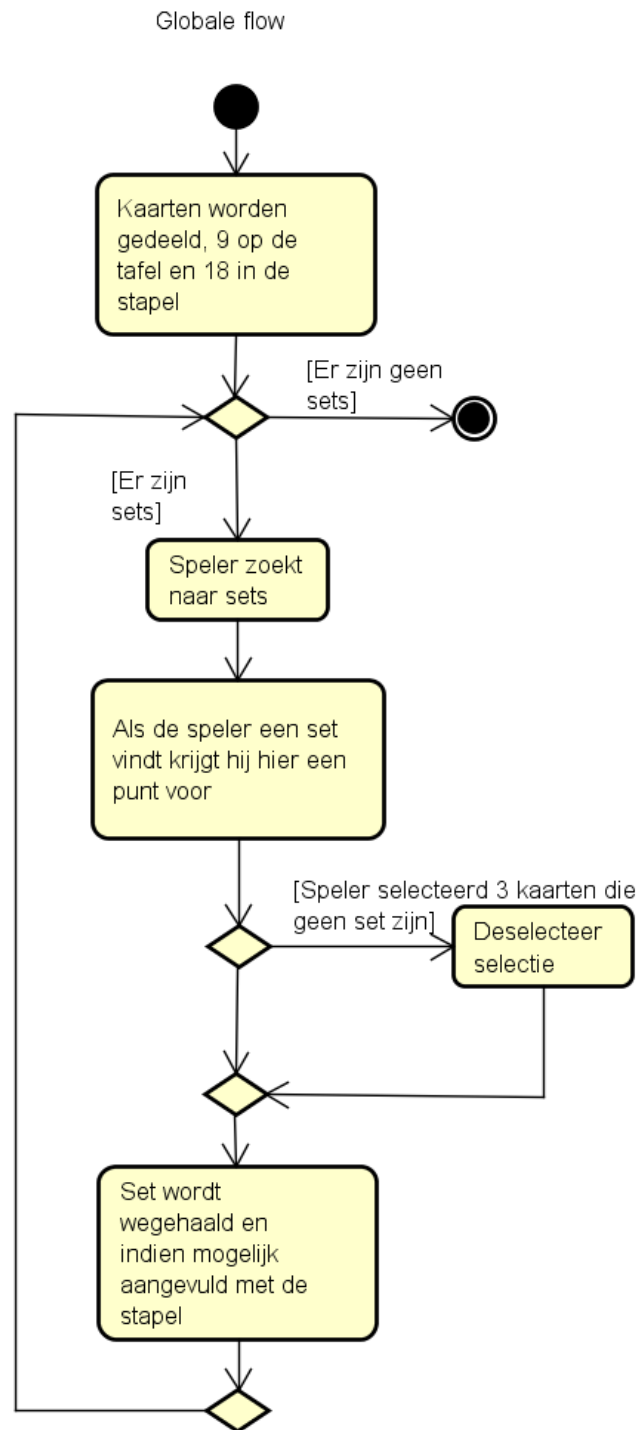
Nummer	Eis	MoSCoW
1.	Er worden 27 unieke kaarten gegenereerd.	M
2.	Een kaart bestaat uit drie attributen, een aantal, een kleur en een vorm.	M
3.	Een set bestaat uit drie kaarten waarvan minimaal één attribuut overkomt met alle kaarten in de set. En een attribuut komt niet gedeeltelijk overeen tussen de kaarten.	M
4.	Er is een stapel met kaarten die nog niet in het spel zijn gebruikt.	M
5.	Er ligt altijd een meervoud van het aantal attributen, met een maximum van het aantal attributen keer 3, aan kaarten open op het scherm.	M
6.	Als er een set wordt gevonden worden deze van de tafel afgehaald en indien mogelijk aangevuld met kaarten van de stapel.	M
7.	Een speler kan tussen de 0 en het aantal attributen aan unieke kaarten selecteren.	M
8.	Als een speler drie kaarten selecteert wordt er gekeken of het een set is. Is dit het geval, is er een set gevonden. Anders wordt de selectie ongedaan gemaakt	M
9.	Er wordt bijgehouden hoeveel sets de speler heeft gevonden.	M
10.	De kaarten die op tafel liggen worden getekend op het scherm.	M
11.	Het gevonden aantal sets is te lezen na afloop van het spel (console/scherm).	M
12.	Het is mogelijk om te zien welke kaarten zijn geselecteerd.	S
13.	Er is altijd bekend hoeveel sets er op de tafel liggen.	S
14.	De kaarten op het scherm hebben het juiste aantal symbolen, in de juiste kleur en het juiste symbool, volgens de attributen van de kaart.	S
15.	Elke kaart kan maar één keer voorkomen in een spel.	S
16.	De kaarten moeten worden geschud.	S
17.	Het spel stopt als er geen sets meer op de tafel liggen.	S
18.	Elke kaart kan precies één keer worden geselecteerd.	C
19.	Er is te zien hoeveel kaarten er nog in de stapel zitten.	W
20.	Er is een knop die 2 kaarten uit een set laat zien. Dit kan tegen puntenaftrek.	W
21.	Er kunnen meerdere spelers tegen elkaar spelen.	W
22.	Er wordt een vierde attribuut toegevoegd. Er komen dan ook 12 kaarten op de tafel en er zijn dan ook 81 kaarten in het spel.	W
23.	Er is een knop die drie extra kaarten op de tafel legt. Dit kan tegen puntenaftrek.	W
24.	Een speler kan een geselecteerde kaart ook de-selecteren.	W

Figuur 1 - eisen

Dit schema kunnen we later gebruiken om het stappenplan op te stellen, te controleren of we alle eisen hebben verwerkt en voor het bepalen hoe we bepaalde functionaliteiten gaan implementeren. Er is gekozen om requirement 19, 20 en 24 meteen toe te voegen. Verder is er bij requirement 11 gekozen om meteen het scherm te gebruiken in plaats van de console. Requirement 23 wordt uitgewerkt als er tijd over is.

3.3. Globale flow

Het volgende diagram laat de simpele flow van het programma zien, dit is vooral bedoeld om te kijken of de opdracht begrepen is.



Figuur 2 - globale flow

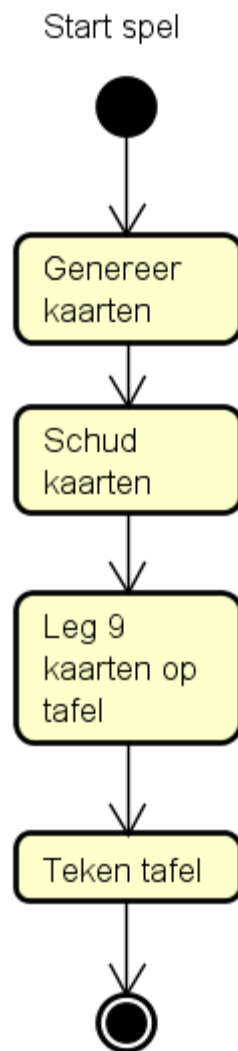
In dit schema is niet opgenomen dat het spel ten alle tijden kan worden gestopt, dit kan uiteraard wel. Verder kan een speler ook opnieuw spelen als het spel is afgelopen. In het volgende hoofdstuk wordt dieper in gegaan op de flow van het programma.

3.4. Flow van het programma

Dit hoofdstuk zal per fase van het spel beschrijven wat het programma doet.

3.4.1. Start van het spel

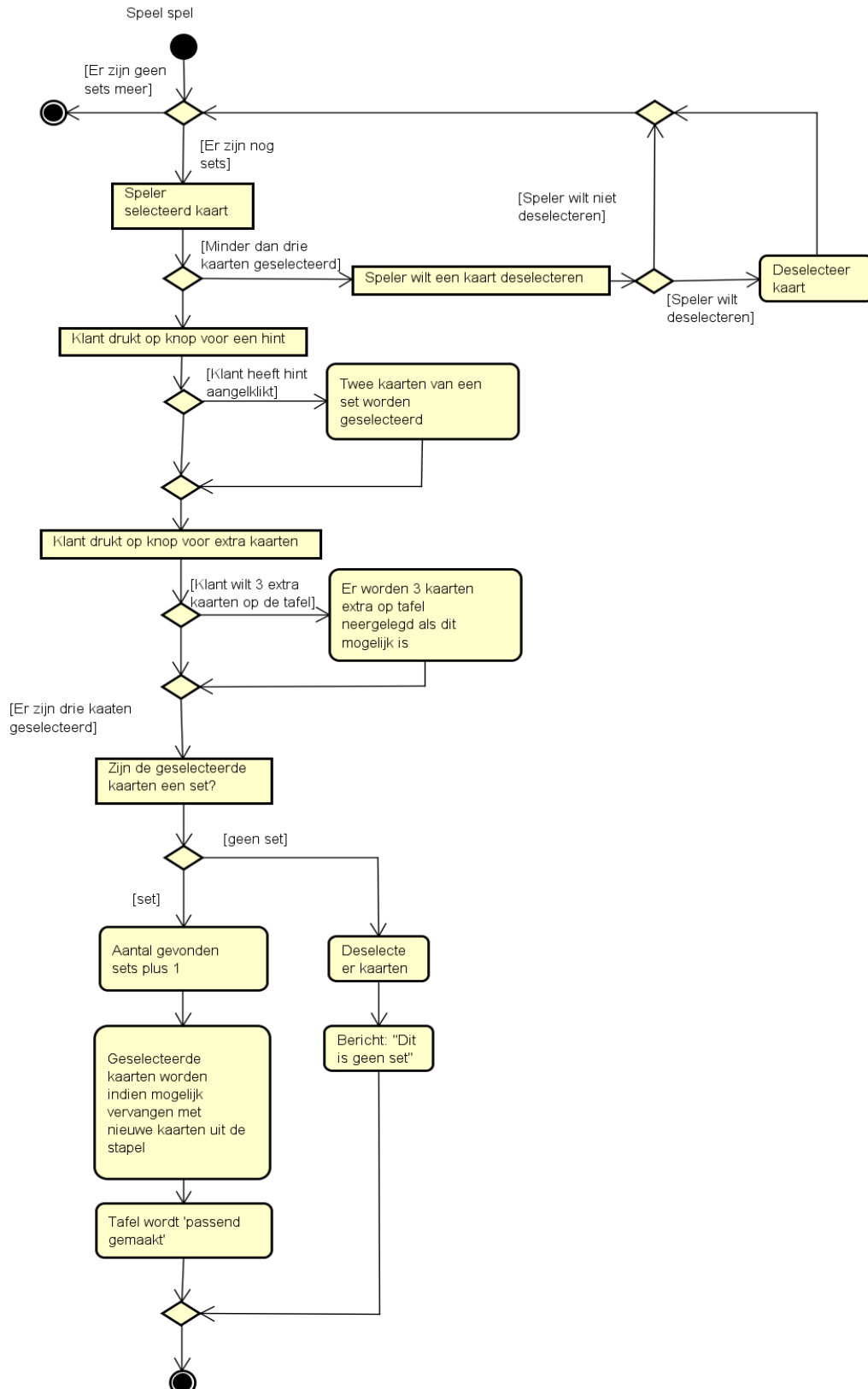
Het volgende diagram beschrijft wat er gebeurt als het spel opstart, of opnieuw wordt gespeeld na een spelletje.



Figuur 3 - start spel

3.4.2. Verloop van een spel

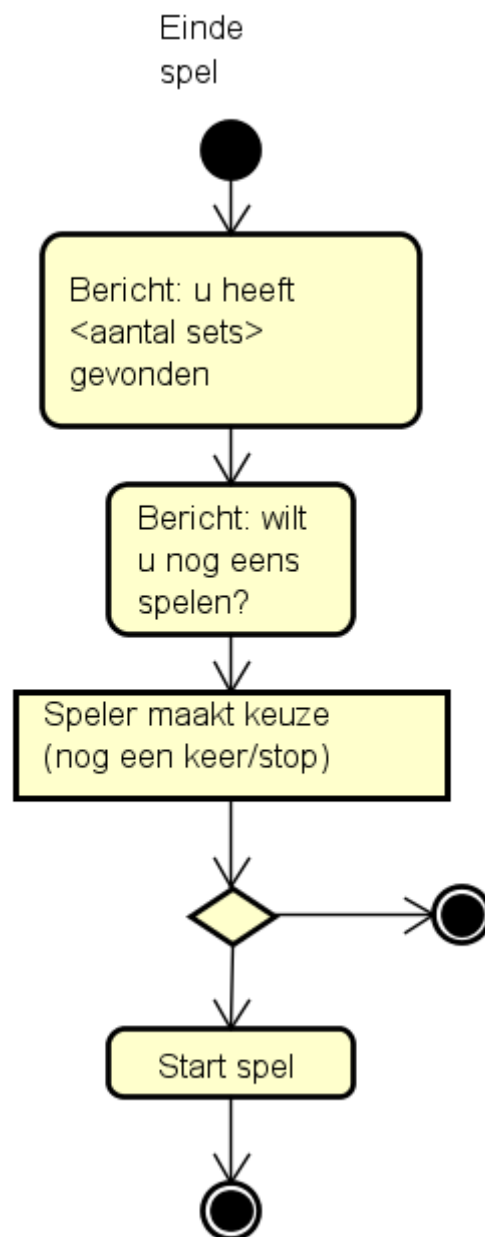
Het volgende diagram weergeeft wat het programma doet tijdens het spel. U ziet dat een speler kaarten kan selecteren, kaarten kan deselecteren, sets kan vinden en een foute set kan vinden. Verder kan het spel eindigen als er geen sets meer op tafel liggen. Als er eenmaal om extra kaarten is gevraagd, blijven er gedurende de rest van het spel extra kaarten liggen.



Figuur 4 - verloop spel

3.4.3. Einde spel

Het volgende diagram is de loop van het programma vlak nadat het spel is afgelopen. Dit zorgt onder andere voor berichtgeving van de score als het maken van een knop om nog een keer te spelen.












Figuur 5 - einde spel

3.5. Schermontwerpen

In dit subhoofdstuk wordt beschreven hoe het eindproduct er ongeveer uit komt te zien, dit is gedaan aan de hand van schetsen. In werkelijkheid zal alles netjes zijn uitgelijnd.

3.5.1. Spel

Dit scherm zullen spelers zien tijdens het spelen van het spel.

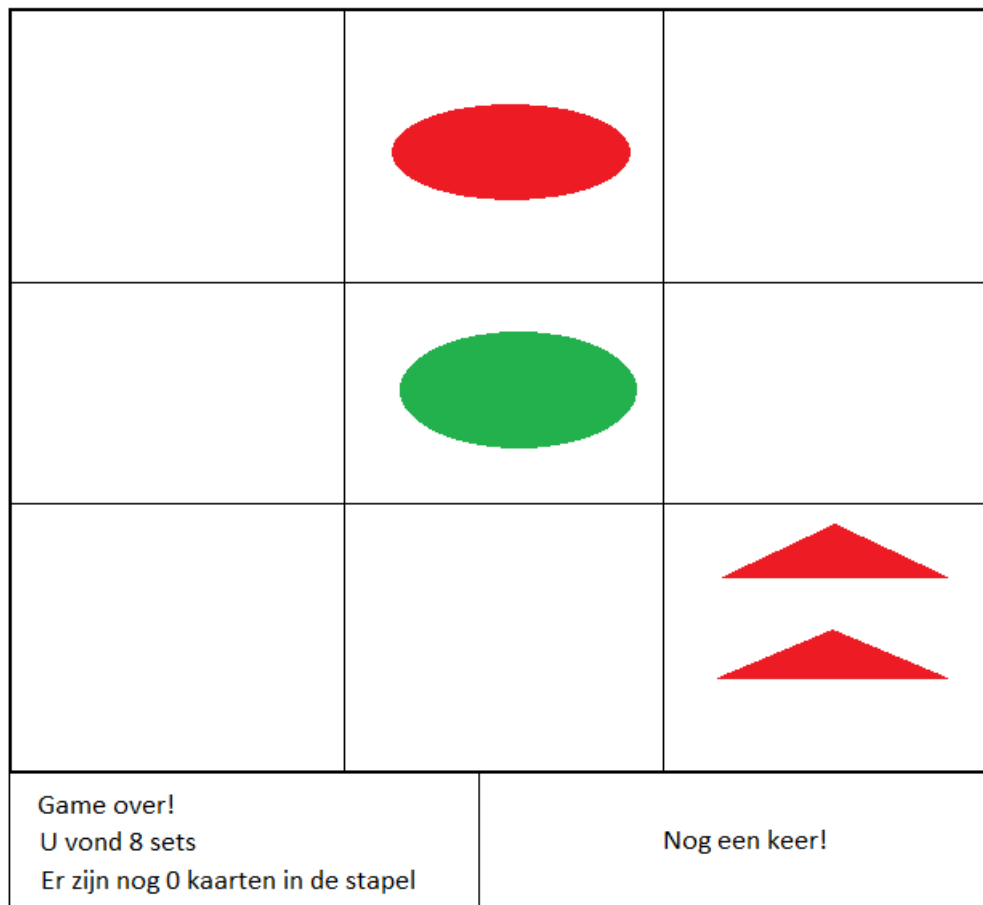
		
		
		
Aantal gevonden sets: 0 Kaarten in de stapel: 18 Sets op tafel: 3		Geef me een hint
		Geef me extra kaarten

Figuur 6 - scherm tijdens spel

De kaarten met een achtergrond kleur zijn geselecteerd.

Einde spel

Dit scherm zal een speler zien als er geen sets meer zijn, het spel is dan voorbij.



Figuur 7 - scherm einde spel

Als er op "Nog een keer!" wordt geklikt, begint er een nieuw spel

4. Technisch ontwerp

In dit hoofdstuk zal in technische termen worden beschreven hoe SubSet zal worden gebouwd. In dit hoofdstuk gaan we het dus hebben over de benodigde variabelen, functies en events. Verder zal een uitgebreidere programma flow aan bod komen.

4.1. Lijst van variabele, functies en events

In de volgende tabel zullen alle functie, variabele en events worden genoemd, waarvan wordt verwacht dat deze gebruikt gaan worden. De namen in deze tabel zijn voorlopig, mocht er blijken dat het duidelijker is om een andere naam te kiezen wordt gedaan. Het kan ook zijn dat er functies of parameters te veel of te weinig zijn, dit zal in het programma worden gewijzigd, maar niet in deze tabel.

Module	Functie/variabele/event	Data-/returntype	Naam	Ingebouwd in processing?
Beroepsproduct	Variabele	Final String[]	EIGENSCHAPPEN	Nee
	Variabele	Final int	VELDZIJDEX	Nee
	Variabele	Int	veldZijdeY	Nee
	Variabele	Final float	FACTOR_HOOGTE_SPEELVELD	Nee
	Variabele	String[]	stapel	Nee
	Variabele	String[]	Tafel	Nee
	Variabele	int	score	Nee
	Variabele	Int	aantalGevondenSets	Nee
	Variabele	float	mouseX	Ja
	Variabele	float	mouseY	Ja
	Variabele	float	width	Ja
	Variabele	float	height	Ja
	Event	void	setup()	Ja
	Event	void	mousePressed()	Ja
	Event	Void	draw()	Ja
	Functie	void	reset()	Nee
	Functie	<x>[]	Subset(<x>, int start, int end)	Ja
Kaarten	Functie	String[]	genereerKaarten()	Nee
	Functie	String[]	schudKaarten(String[] kaarten)	Nee
	Functie	float	pow(float a, float b)	Ja
	Functie	Int	Floor(float a)	Ja
	Functie	<x>[]	append(<x>[] a, <x> waarde)	Ja
	Functie	String	.substring(int start, int end)	Ja
	Functie	boolean	.equals(Object)	Ja
	Variabele	Int	<x>[].length	Ja
	Functie	int	String.length()	Ja
	Functie	Float	Random(float min, float max)	Ja
	Functie	boolean	isSet(final String[] kaarten)	Nee
	Functie	int	aantalSetsOpTafel()	Nee
	Functie	void	selecteerTweeKaartenVanEenSet()	Nee

	Functie	boolean	zijnKaartenValide(final String[] kaarten)	Nee
	Functie	String[]	haalLegeStringsUitArray(final String[] tafel)	Nee
Math	Functie	float[]	positieNaarCoordinaten(final int positie, final int posX, final int posY, final float breedte, final float hoogte)	Nee
	Functie	Int	aantalTrue(final Boolean[] array)	Nee
	Functie	Int[]	indexesTrue(final Boolean[] array)	Nee
Tekenen	Functie	void	Fill(color)	Ja
	Functie	Void	Rect(float x, float y, float width, float height)	Ja
	Functie	Void	Ellipse(float x, float y, float width, float height)	Ja
	Functie	Void	Triangle(float x1, float y1, float x2, float y2, float x3, float y3)	Ja
	Functie	Void	Text(String text, float x, float y)	Ja
	Functie	Void	tekenKaarten(final String[] kaarten, final int posX, final int posY, final int breedte, final int hoogte)	Nee
	Functie	Void	tekenKaart(final String eigenschappen, final int positie, final int posX, final int posY, final int breedte, final int hoogte, final boolean isGeselecteerd)	Nee
	Functie	Void	setSymbolKleur(final String kleurEigenschap)	Nee
	Functie	float	bepaalYSymbool(final float y, final int nummerSymbool, final int aantal, final float kaartHoogte, final float symboolHoogte)	Nee
	Functie	Void	tekenSymbolen(final String aantalSymbolen, final Strig vorm, final float kaartHoogte, final float kaartBreedte, final int[] pos)	Nee
	Functie	Void	tekenScore(final int posX, final int posY)	Nee
	Functie	Void	tekenBord(final int posX, final int posY, final int breedte, final int hoogte)	Nee
	Functie	Void	tekenGameOver(final int posX, final int posY, final int breedte)	Nee
	Functie	Void	tekenAlles()	Nee
	Functie	Void	tekenFooter(final int posX, final int posY, final int breedte)	Nee
	Functie	Void	setTextGrootte(final int textGrootte, final int breedte)	Nee
User Input	Functie	boolean	muisInVierkant(final int x, final int y, final int breedte, final int hoogte)	Nee

Functie	Boolean[]	deselecteerAlles()	Nee
Functie	Boolean[]	VerwerkKlik(final int pos, final int posY, final int breedte, final int hoogte)	Nee
Functie	String[]	bepaalGeselecteerdeKaarten()	Nee
Functie	Void	vervangSetMetNieuweKaarten()	Nee

Figuur 8 – Functies en variabele

4.2. Lijst van te maken globale variabelen:

Voor de volgende variabelen geldt dat ze worden aangeroepen in functies die in zowel setup als mousePressed worden aangeroepen. Het is dus nodig dat deze globaal zijn, omdat een event niet geparametiseerd kan worden.

Datatype	Naam	Uitleg
Final String[]	EIGENSCHAPPEN	Wordt gebruikt om de eigenschappen en variaties van eigenschappen die de kaarten kunnen hebben bij te houden.
Final float	FACTOR_HOOGTE_SPEELVELD	Wordt gebruikt om de hoogte van het gedeelte voor de footer en voor de kaarten uit te rekenen.
Final int	VELDZIJDEX	Hoeveel kaarten er in de x richting op de tafel kunnen liggen.
Int	veldZijdeY	Hoeveel kaarten er in de y richting op de tafel kunnen liggen.
String[]	stapel	Kaarten die nog gebruikt kunnen worden.
String[]	tafel	Kaarten die op tafel liggen.
Int	aantalGevondenSets	Score van de speler.
int	score	Score van de speler, houdt rekening met min punten voor hints.

Figuur 9 – globale variabele

4.3. Lijst van te maken functies:

In dit subhoofdstuk wordt wat dieper ingegaan op wat elke functie doet, en wat de parameters verwachten. Alleen de functies die zelf geschreven moeten worden, worden behandeld. Er zijn twee voordelen om dit zo uitgebreid te doen: er ontstaat een beter beeld hoe het programma gemaakt gaat worden. En er wordt een fundering gelegd voor de toekomstige documentatie.

Functie en functie uitleg	Uitleg return waarde	Parameter	Uitleg parameter
<code>void reset()</code> , reset het hele spel.	-	-	-
<code>String[] genereerKaarten()</code> , genereerd alle combinaties van EIGENSCHAPPEN, en zo 27 kaarten in het standard spel.	Geeft een array van strings terug die kaarten representeren.	-	-
<code>String[] schudKaarten(String[] kaarten)</code> , door een groot aantal keer wisselen van twee willekeurige kaarten worden de kaarten geschud.	Geeft de geschudde kaarten terug.	<code>Final String[] kaarten</code>	Kaarten die worden geschud.
<code>Void selecteerTweeKaartenVanEenSet()</code> , deselecteerd alle geselecteerde kaarten en selecteerd 2 kaarten van een set	-	-	-
<code>Boolean zijnKaartenValide(final String[] kaarten)</code> , geeft terug of de array alleen valide karten bevat, en of het kaarten bevat.	Als true, kaarten bevat ten minste 1 kaart die eigenschappen bezit en geen kaart(en) die geen eigenschappen bevatten	<code>String[] kaarten</code>	Kaarten die worden gecontroleerd op validiteit.
<code>Float[] positieNaarCoordinaten(final int positie, final int posX, final int posY, final float breedte, final float hoogte)</code> , zet de index van een kaart op de tafel om naar een array van 2 lang met en x en een y coördinaat.	X- en y-coördinaten van de tekenen kaart.	<code>Final int positie</code>	Index van een kaart op de tafel
		<code>Final int posX</code>	X positie vanaf waar kaarten mogen worden getekend.
		<code>Final int posY</code>	Y positie vanaf waar kaarten mogen worden getekend.
		<code>Final float breedte</code>	De breedte waarbinnen alle

			kaarten moeten worden getekend.
		Final float hoogte	De hoogte waarbinnen alle kaarten moeten worden getekend.
		Final float	
Boolean isSet (final String[] kaarten), kijkt of de kaarten een set vormen	True als alle kaarten gezamenlijke eigenschap(pen) hebben en/of eigenschap(pen) die over alle kaarten variëren, anders false.	Final String[] kaarten	Kaarten waarvan wordt bekeken of het een set is.
Int aantalSetsOpTafel (), berekend het aantal sets dat op tafel ligt. Dit gebeurt door door alle combinaties van 3 in 'tafel' heen te itereren en hierop isSet () toe te passen	Het aantal sets dat op een willekeurig moment op tafel ligt	-	-
Int aantalTrue (final boolean[] array), geeft het aantal keer dat true in de array staat	Het aantal keer dat true voor komt in een boolean array	Final boolean[]	De array waarvan het aantal keer true wordt uitgerekend.
Int[] indexesTrue (final boolean[]), geeft de indexes waar true in staat.	Geeft een array van indexes waarop true staat in de boolean array	Final boolean[]	De array waarvan de indexes van true wordt bepaalt.
Void tekenKaarten (final String[] kaarten, final int posX, final int posY, final int breedte, final int hoogte), tekent alle kaarten die op tafel liggen op het scherm.	-	Final String[] kaarten	Kaarten die worden getekend.
		Final int posX	X-positie vanaf waar wordt getekend.
		Final int posY	Y-positie vanaf waar wordt getekend.
		Final int breedte	Hoe breed er wordt getekend.
		Final int hoogte	Hoe hoog er wordt getekend.
Void tekenKaart (final String eigenschappen, final int positie, final int posX, final int posY, final int breedte, final int hoogte, final	-	Final String eigenschappen	De eigenschappen die deze kaart bevat, aantal symbolen, kleur

boolean isGeselecteerd), tekend één kaart.			symbool en vorm symbool.
		Final int positie	De positie van de kaart op de tafel. De tafel is als volgende genummerd: 0, 1, 2 3, 4, 5 6, 7, 8.
		Final int posX	X positie vanaf waar alle kaarten moeten worden getekend.
		Final int posY	Y positie vanaf waar alle kaarten moeten worden getekend.
		Final int breedte	Breedte waarbinnen alle kaarten moeten worden getekend.
		Final int hoogte	Hoogte waarbinnen alle kaarten moeten worden getekend.
		Final boolean isGeselecteerd	Of de kaart is geselecteerd, als dit zo is wordt dit zichtbaar gemaakt.
Void setSymbolKleur (final String kleurEigenschap), bepaald de fill-kleur.	-	Final String kleurEigenschap	Karakter dat een kleur weergeeft, allen r, g, b zijn gedefinieerd.
Float bepaalYSymbol (final float y, final float nummerSymbool, final int aantal, final float kaartHoogte, final float symboolHoogte), geeft de y positie terug voor het symbool.	De y positie voor het symbool waarvoor deze parameters gelden.	Final float y	De y-positie van de kaart die waarvoor deze symbolen zijn.
		Final int nummerSymbool	Het nummer van het symbool. Is nodig als er meer dan één symbool kan zijn per kaart, wordt gebruikt voor de relatieve positionering van het symbool.

		Final int aantal	Het aantal symbolen dat er op de kaart staan, wordt in combinatie met 'nummerSymbool' gebruikt voor de relatieve positionering van het symbool.
		Final float kaartHoogte	De hoogte van deze kaart, wordt gebruikt om de symbolen netjes te verdelen over de kaart.
		Final float symboolHoogte	Hoogte van één symbool.
Void tekenScore (final int posX, final int posY), tekend de score onder in beeld. Gaat ervan uit dat hij mag door tekenen totdat de onderkant van het scherm is bereikt.	-	Final int posX	X-positie om vanaf te tekenen.
		Final int posY	Y-positie om vanaf te tekenen.
void tekenBord (final int posX, final int posY, final int breedte, final int hoogte), tekend alles wat er op het scherm te zien is.	-	Final int posX	X-positie om vanaf te tekenen.
		Final int posY	Y-positie om vanaf te tekenen.
		Final int breedte	Hoe breed er wordt getekend.
		Final int hoogte	Hoe hoog er wordt getekend.
Void tekenGameOver (final int posX, final int posY, final int breedte), tekend hetgeen dat te zien is als het spel is afgelopen.	-	Final int posX	X-positie vanaf waar wordt getekend.
		Final int posY	Y-positie vanaf waar wordt getekend.
		Final int breedte	Breedte van het gedeelte waar GameOver wordt getekend.
Void tekenAlles (), tekent alles, wrapper voor tekenBord, voorkomt extra globale variabelen.	-	-	-

Void tekenFooter (final int posX, final int posY, final int breedte), tekend ofwel GameOver, ofwel score. Dit gebeurt op basis van 'speel'	-	Final int posX	X-positie vanaf waar wordt getekend.
		Final int posY	Y-positie vanaf waar wordt getekend.
		Final int breedte	Breedte van de footer.
Boolean muIsInVierkant (final int x, final int y, final int breedte, final int hoogte), geeft terug of de muis in het vierkant zit.	True als de muis zich in het gespecificeerde vierkant bevind, anders false.	Final int x	X-positie van het vierkant.
		Final int y	Y-positie van het vierkant.
		Final int breedte	Breedte van het vierkant.
		Final int hoogte	Hoogte van het vierkant.
Boolean[] deselecteerAlles (), zet alles in de 'selectie' array naar false.	Nieuwe array van selectie.		
Boolean[] VerwerkKlik (final int posX, final int posY, final int breedte, final int hoogte), Als erop een kaart is geklikt wordt deze geselecteerd/gedeselecteerd. Selecteren kan alleen als er minder dan 3 kaarten in de 'selectie' true zijn. Als er 3 kaarten in de 'selectie' zitten wordt er gekeken of dit een set is, 'vervangSetMetNieuweKaarten', anders 'deselecteerAlles'	Nieuwe array van selectie.	Final int posX	X-positie vanaf waar de kaarten worden getekend.
		Final int posY	Y-positie vanaf waar de kaarten worden getekend.
		Final int breedte	Breedte waar de kaarten in worden getekend.
		Final int hoogte	Hoogte waar de kaarten in worden getekend
String[] bepaalGeselecteerdeKaarten (), kijkt welke kaarten er zijn geselecteerd op basis van 'tafel' en 'selectie'	Kaarten die zijn geselecteerd.	-	-

Void vervangSetMetNieuweKaarten() , vervangt indien mogelijk de gevonden set met nieuwe kaarten van de stapel. Maakt gebruik van 'stapel' en 'tafel'.	-	-	-
String[] haalLegeStringsUitArray(final String[] array), haalt alle lege strings uit de array, wordt gebruikt om de array van tafel in te korten als er invalide kaarten in zitten.	Nieuwe array zonder lege strings	Final String[] array	Array waaruit lege strings worden verwijderd.
Void setTextSize(final int textGrootte, final int breedte), roept textSize() aan, met een constrain zodat de text binnen de aangegeven breedte blijft. Is in een functie gezet, omdat gameOver en tekenScore dezelfde tekst grootte nodig hebben.	-	Final int textGrootte	TextGrootte die vooraf wordt uitegerekend voor de hoogte
		Final int breedte	Breedte waar de tekst binnen moet blijven.

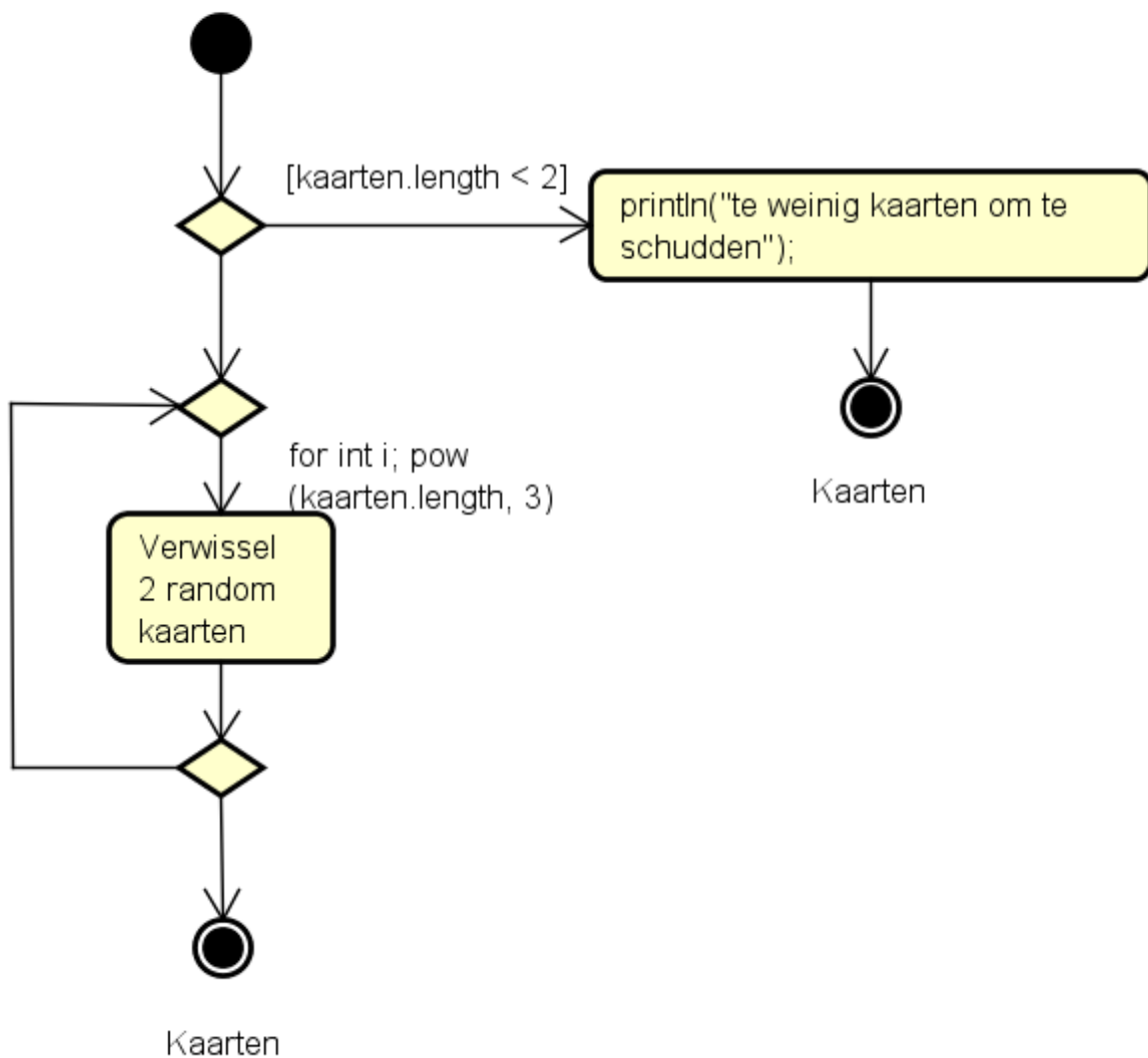
Figuur 10 - functies

4.4. Activity diagrammen

In dit subhoofdstuk wordt de flow van de functies toegelicht, dit wordt gedaan met behulp van een activity diagram. Er is gekozen om alleen functies die keuzemomenten hebben toe te lichten, verder zijn alleen functies toegelicht waarvan al een goed beeld bestaat hoe ze geïmplementeerd gaan worden. Ook wel bekend als flow chart. In deze diagrammen staat '<>' voor een op te zoeken waarde, en een '/' betekend een uit te rekenen waarde.

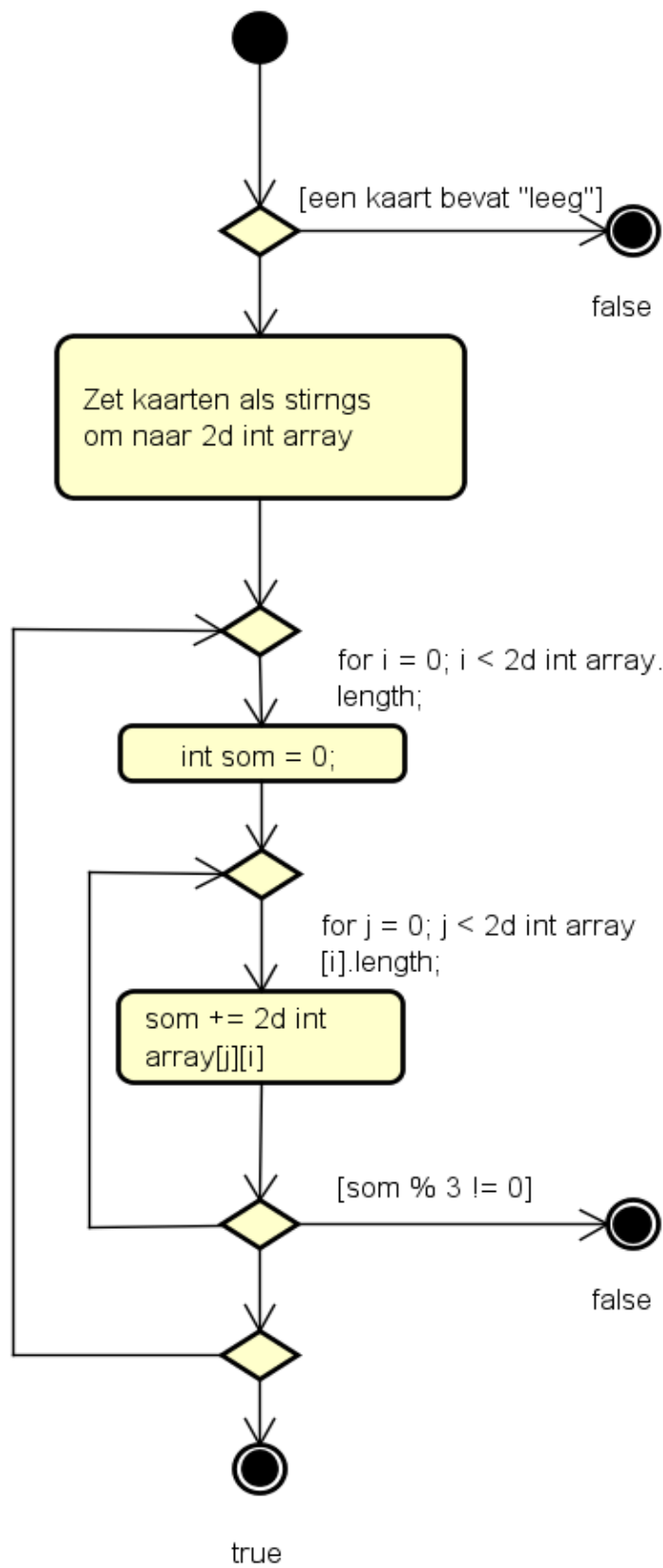
4.4.1. schudKaarten

String[] kaarten



Figuur 11 – schud kaarten

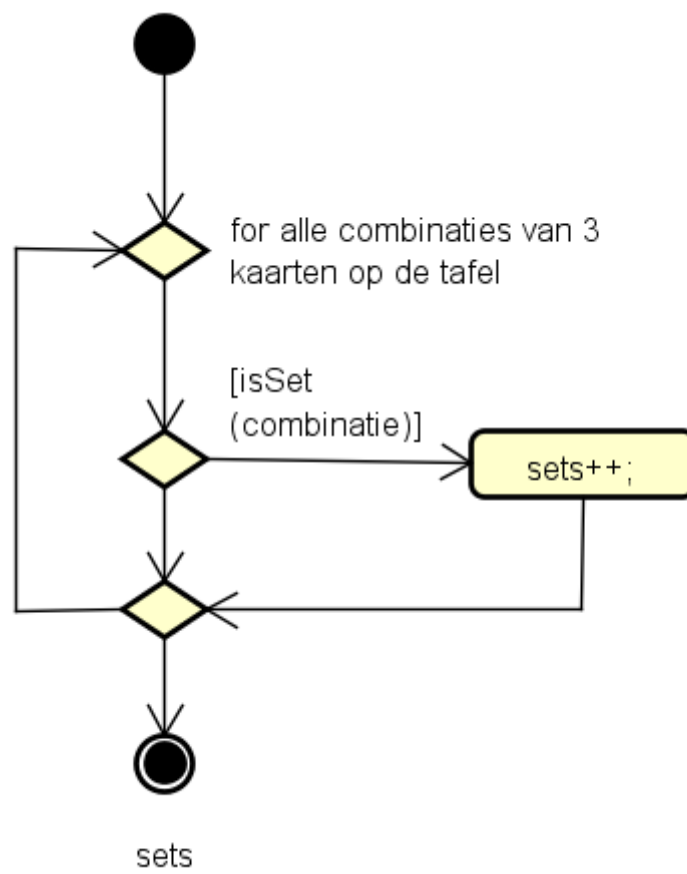
4.4.2. isSet



Figuur 12 - isSet

Opmerking: er is gekozen om ipv. "leeg" een lege string ("") te gebruiken.

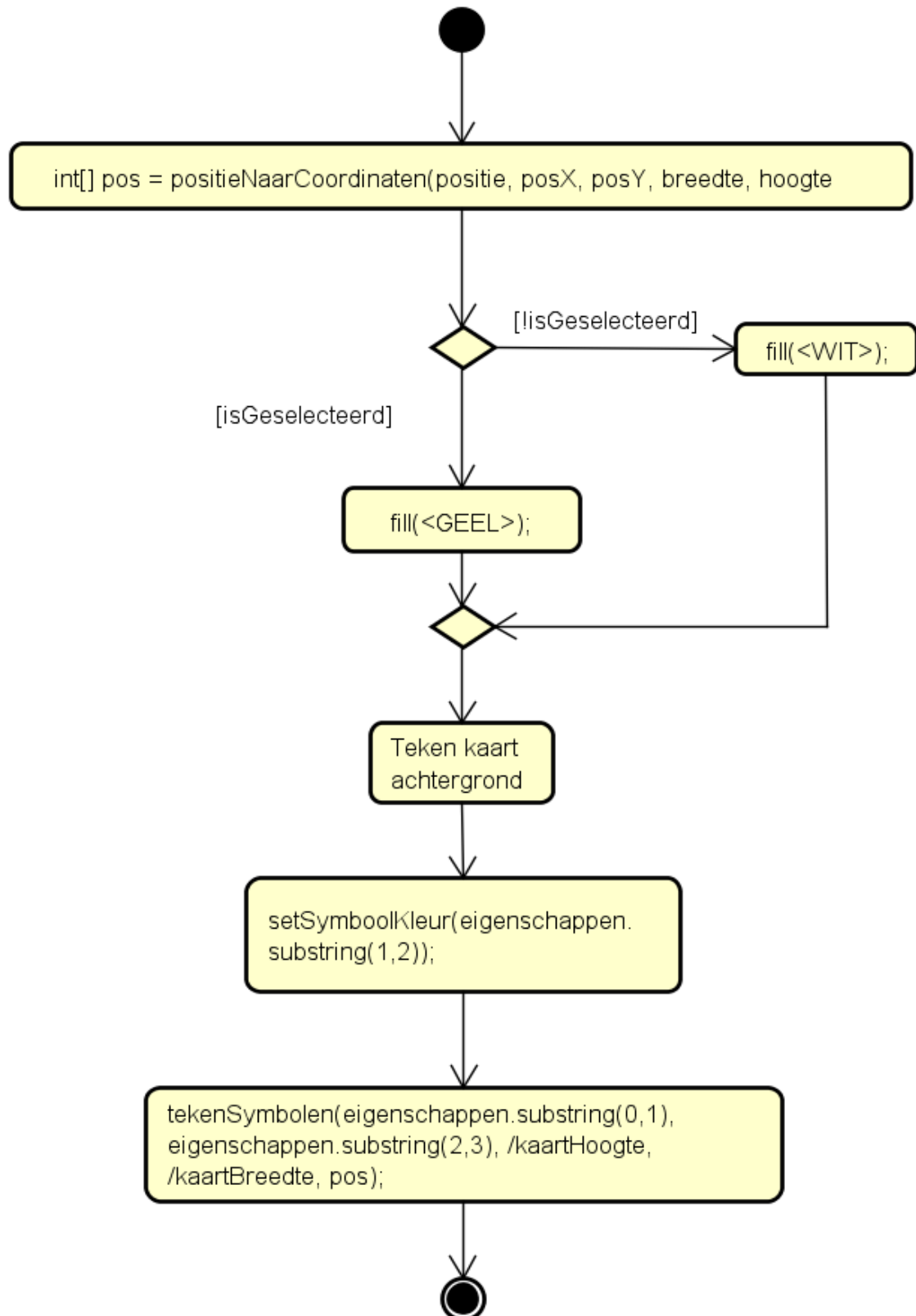
4.4.3. aantalSetsOpTafel



Figuur 13 - aantal sets op tafel

4.4.4. tekenKaart

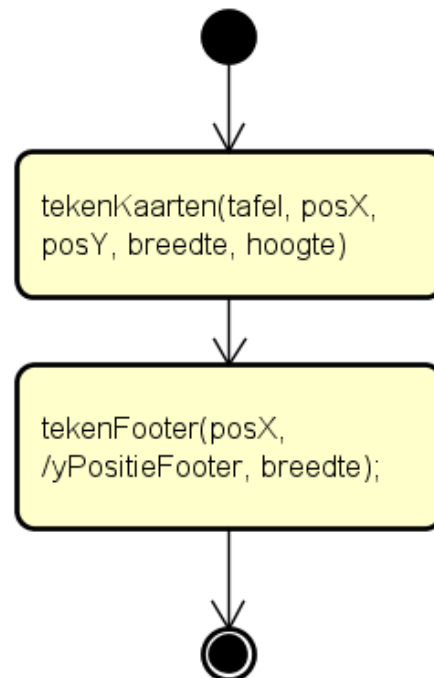
final String eigenschappen, final int positie, final int posX, final int posY, final int breedte, final int hoogte, final boolean isGeselecteerd



Figuur 14 - teken kaart

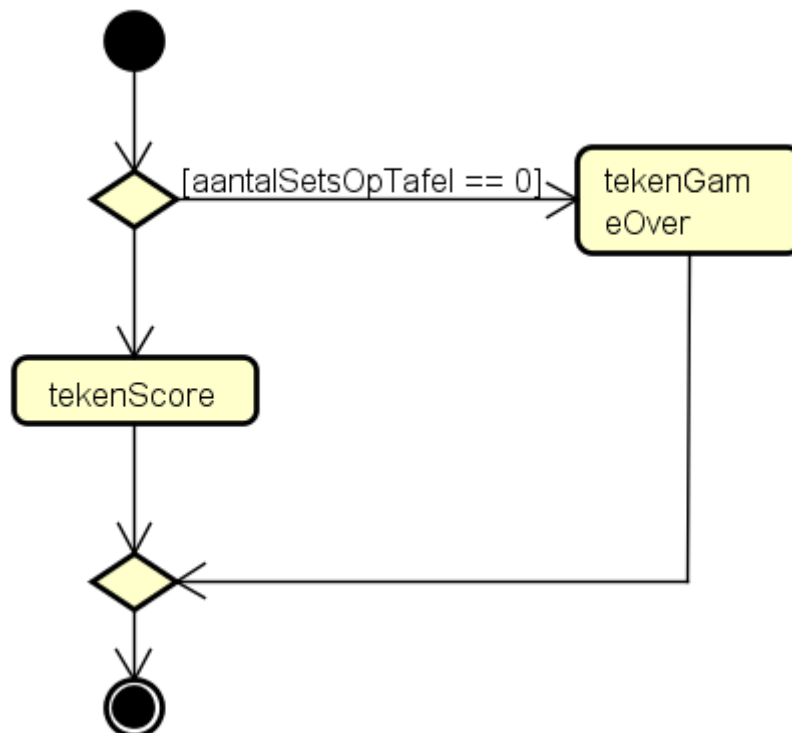
4.4.5. tekenBord

final int posX, final int posY,
final int breedte, final int
hoogte



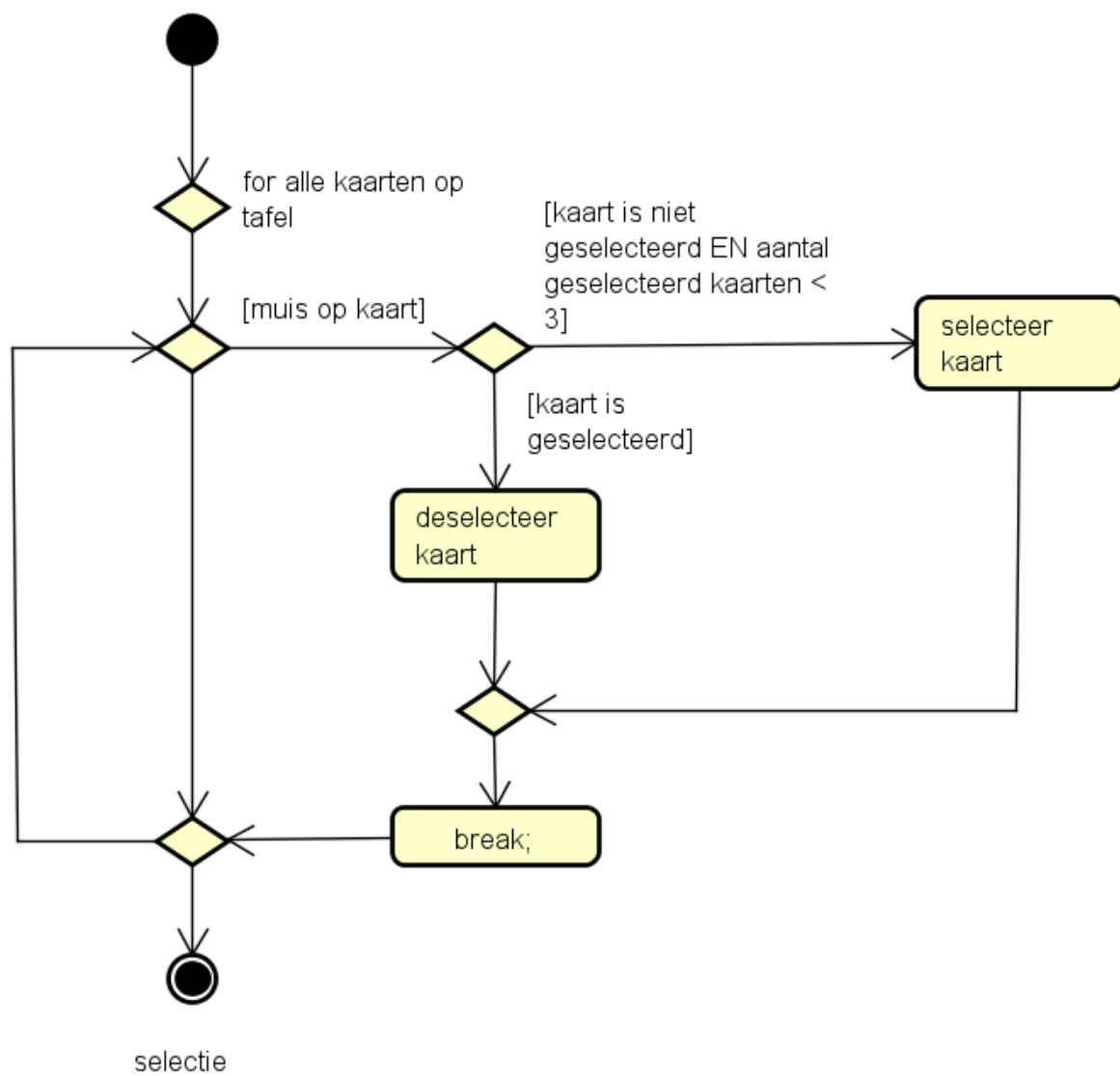
Figuur 15 - teken bord

4.4.6. tekenFooter



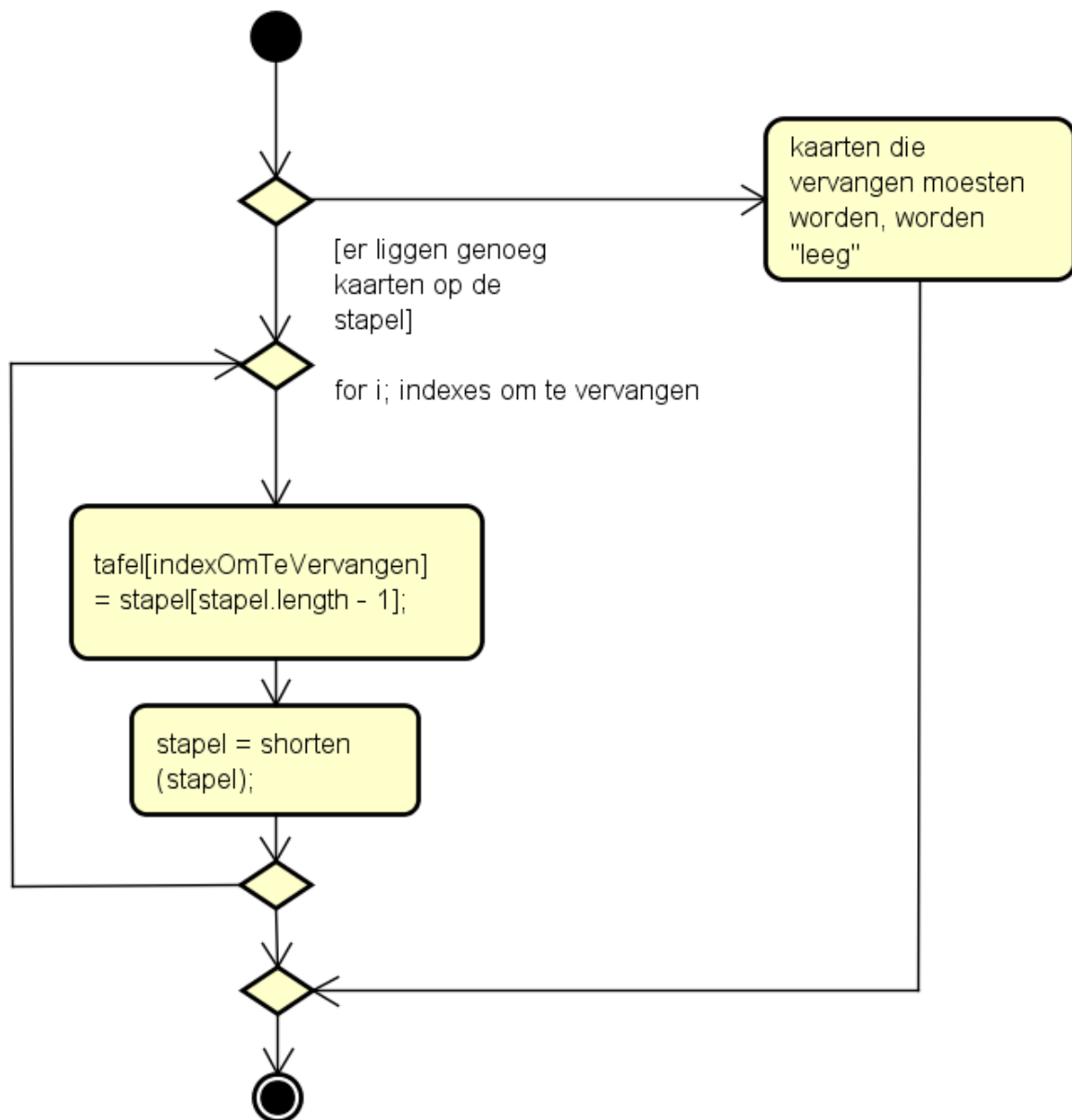
Figuur 16 - teken footer

4.4.7. verwerkKlik



Figuur 17 - verwerk klik

4.4.8. vervangSetMetNieuweKaarten



Figuur 18 - vervang set met nieuwe kaarten

5. Test plan

In dit hoofdstuk is opgenomen wat er getest gaat worden, of bepaalde functionaliteiten werken, eventuele oplossingen voor problemen, en als dit zo is hoe de nieuwe test verloopt. Er is getest door de tafel en stapel hard geprogrammeerde waarden te geven, maar ook zonder dit te doen.

Test nummer	Te testen functionaliteit	Invoer	Verwacht resultaat	Verkregen resultaat	Oplossing	Nieuw resultaat
1.	Selectie en sets	-	-	-	-	
1.1	Wordt er bij het juiste aantal kaarten gekeken of er een set is.	1 kaart selecteren.	Kaart krijgt een gele achtergrond. (wordt geselecteerd)	Kaart krijgt een gele achtergrond.	-	Kaart krijgt een gele achtergrond.
		2 kaarten selecteren.	Kaarten krijgen een gele achtergrond.(worden geselecteerd)	Kaarten krijgen een gele achtergrond.	-	Kaarten krijgen een gele achtergrond.
		Kaart die al geselecteerd is nog een keer aanklikken.	Kaart krijgt een witte achtergrond. (worden gedeselecteerd)	Kaart krijgt een witte achtergrond.	-	Kaart krijgt een witte achtergrond.
1.2	Is de controle op sets juist.	3 kaarten selecteren met eigenschappen die voor 2 kaarten gelden.	Het is geen set, kaarten worden gedeselecteerd.	Kaarten worden gedeselecteerd.	-	Kaarten worden gedeselecteerd.
		3 kaarten selecteren met een gezamenlijke kleur, vorm en verschillende aantallen.	Het is een set, kaarten worden vervangen. Score wordt verhoogt.	Kaarten worden vervangen, score wordt verhoogt.	-	Kaarten worden vervangen, score wordt verhoogt.
		3 kaarten selecteren met een gezamenlijke vorm, aantallen en verschillende kleuren.	Het is een set, kaarten worden vervangen.	Kaarten worden vervangen, score wordt verhoogt.	-	Kaarten worden vervangen, score wordt verhoogt.
		3 kaarten selecteren met verschillende kleuren, vormen en aantallen	Het is een set, kaarten worden vervangen.	Kaarten worden vervangen, score wordt verhoogt.	-	Kaarten worden vervangen, score wordt verhoogt.

		Aantal sets op tafel tellen en vergelijken met het aantal sets dat in de footer staat.	Het aantal sets komt overeen.	Het aantal sets komt overeen.	-	Het aantal sets komt overeen.
1.3	Wordt de tafel aangevuld?	3 kaarten, die een set vormen, selecteren, stapel is 3 of groter.	Kaarten worden gedeselecteerd en vervangen door nieuwe kaarten. Aantal gevonden sets wordt opgehoogd.	Kaarten worden vervangen.	-	Kaarten worden vervangen.
		3 kaarten, die een set vormen, selecteren, stapel is 0 hoog.	Kaarten worden weggehaald. Aantal gevonden sets wordt opgehoogd.	Index out of bounds error in isSet	If statement die kijkt of een kaart "leeg" bevat en als dat zo is false returnt.	Kaarten worden weggehaald
2.1	Gameover.	Tafel leeg maken totdat er geen set meer ligt.	Game over.	Game over footer verschijnt, input op tafel wordt geblokkeerd.	-	Game over footer verschijnt, input op tafel wordt geblokkeerd.
2.2	Opnieuw spelen.	Druk na game over op de knop opnieuw spelen.	Spel begint opnieuw.	Spel start opnieuw.	-	Spel start opnieuw.
3.1	Hint	Druk op “geef me een hint”	Er worden twee kaarten van een set geselecteerd. Score wordt verlaagt met 1.	Er worden twee kaarten van een set geselecteerd. Score wordt verlaagt met 1.	-	Er worden twee kaarten van een set geselecteerd. Score wordt verlaagt met 1.
3.2		Druk op “geef me een hint” als er nog maar 1 set op tafel ligt.	Er worden twee kaarten van de set geselecteerd. Score wordt verlaagt met 1.	Er worden twee kaarten van de set geselecteerd. Score wordt verlaagt met 1.	-	Er worden twee kaarten van de set geselecteerd. Score wordt verlaagt met 1.
3.3	Nieuwe kaarten	Druk op “geef me extra kaarten”	Er komen 3 nieuwe kaarten aan de rechterkant van het veld. Score gaat 3 omlaag.	Kaarten komen door elkaar te liggen, wel drie extra.	veldZijdeX en VELDZIJDEY waren omgedraaid in positieNaarCoordinaten.	Er komen 3 nieuwe kaarten aan de rechterkant van het veld.
3.4		Druk op “geef me extra kaarten” als er minder dan 3	Er gebeurt niets.	Er gebeurt niets.	-	Er gebeurt niets.

	kaarten op de stapel liggen.				
	Leg extra alle verschillende mogelijkheden van extra kaarten op tafel, en klik op een kaart.	Aangeklikte kaart wordt geselecteerd.	Aangeklikte kaart wordt geselecteerd.	-	Aangeklikte kaart wordt geselecteerd.
	Selecteer een set als de stapel 0 kaarten bevat, en er extra kaarten op de tafel liggen.	De kaarten worden groter en vullen het hele veld.	De kaarten worden groter en vullen het hele veld.	-	De kaarten worden groter en vullen het hele veld.
	Herstart een spel als er extra kaarten op de tafel hebben gelegen.	Spel start normaal op met 3 x 3 kaarten.	Spel start normaal op met 3 x 3 kaarten.	-	Spel start normaal op met 3 x 3 kaarten.

Figuur 19 - test plan

Uiteraard is dit niet de eerste test, in werkelijkheid is er elke keer dat er een stukje is gebouwd ook getest of dit werkte. Verder is de gebouwde software aan een aantal mensen gegeven met de opdracht “probeer het stuk te maken”. Dit is niet gelukt.

6. Conclusie

Zonder al te veel problemen is het gelukt het spel subset te bouwen binnen de afgesproken periode. Achteraf gezien had er beter gekozen kunnen worden om de kaarten te weergeven als cijfers in plaats van strings, dat had een conversie kunnen schelen (om te bepalen of iets een set is). Er is gekozen om dit niet meer te veranderen, omdat dit zou vereisen dat er een aanzienlijk deel van het programma wordt herschreven. Hier zou voor de gebruiker geen voordeel te halen zijn. Er is gekozen om ervoor te zorgen dat de kaarten en de footer altijd het hele scherm opvullen. Dit houdt dus in dat de kaarten groter worden naar mate er minder kaarten op de tafel liggen. Doordat er nog tijd over was tot de deadline is requirement 23 ook geïmplementeerd.

7. Figuren

Figuur 1 - eisen	5
Figuur 2 - globale flow	6
Figuur 3 - start spel	7
Figuur 4 - verloop spel	8
Figuur 5 - einde spel	9
Figuur 6 - scherm tijdens spel	10
Figuur 7 - scherm einde spel	11
Figuur 8 – Functies en variabele	14
Figuur 9 – globale variabele	14
Figuur 10 - functies	20
Figuur 11 – schud kaarten	21
Figuur 12 - isSet	22
Figuur 13 - aantal sets op tafel	23
Figuur 15 - teken kaart	24
Figuur 16 - teken bord	25
Figuur 17 - teken footer	25
Figuur 18 - verwerk klik	26
Figuur 19 - vervang set met nieuwe kaarten	27
Figuur 20 - test plan	30

8. Bibliografie

HAN. (2017). SPD - Beroepsproduct. *Opdracht beroepsproduct: subset*. Nederland.