

Hogeschool van Arnhem en Nijmegen

Ontwerp Schuifpuzzel

Functioneel en technisch ontwerp van het schuifpuzzelspel

Misja Nabben

10-8-2017

Versie 1.2

1. Contents

1. Contents	1
2. Opdrachtoomschrijving	2
3. Globale flow	3
4. Schermontwerpen.....	4
Start spel.....	4
Kies spelvariant.....	4
Verplaats puzzel	5
Spel uitgespeeld	5
5. Softwarebeschrijving.....	6
Globale softwarestructuur	6
Events, Methoden en Variabelen.....	7
Samenhang schermontwerp en variabelen	9
Programma flow	10
6. Testplan	11

2. Opdrachtomschrijving

De opdracht betreft het realiseren van een schuifpuzzel. De schuifpuzzel moet een spelversie worden van de houten schuifpuzzel. De schuifpuzzel kan bijvoorbeeld bestaan uit 15 puzzelstukjes. Ieder puzzelstukje bevat een getal van 1 tot en met 15. Deze puzzelstukjes worden in een vierkant gelegd waardoor 1 vakje open blijft.

De schuifpuzzel moet door de speler worden gerangschikt op volgorde van de nummering. De puzzelstukjes moeten echter binnen het vierkant blijven. Alleen de stukjes die met een zijde grenzen aan het open vakje kunnen geschoven worden naar dit open vakje. Uiteindelijk is het spel afgelopen als de puzzels op volgorde liggen van linksboven naar rechtsonder. Het aantal verschuivingen die nodig waren wordt geteld en aan de speler getoond.

De volgende minimale eisen ten aanzien van het spel gelden:

- Het moet speelbaar zijn in zowel de 8 (een 3 bij 3) als 15 (een 4 bij 4) schuifjes variant
- De stand van de puzzel moet opgeslagen kunnen worden, waarbij de opgeslagen stand bij het starten weer worden ingelezen;
- Het aantal verschuivingen moet worden vastgelegd

Het spel kan in de toekomst mogelijk uitgebreid worden met de volgende extra functionaliteiten:

1. Een 'random' gehusselde puzzelstart (afhankelijk van de hussel kan het spel mogelijk niet worden uitgespeeld).
2. Het schuiven van meerdere puzzelstukjes tegelijk (in dezelfde rij of kolom);
3. Meer mogelijke puzzelgroottes;

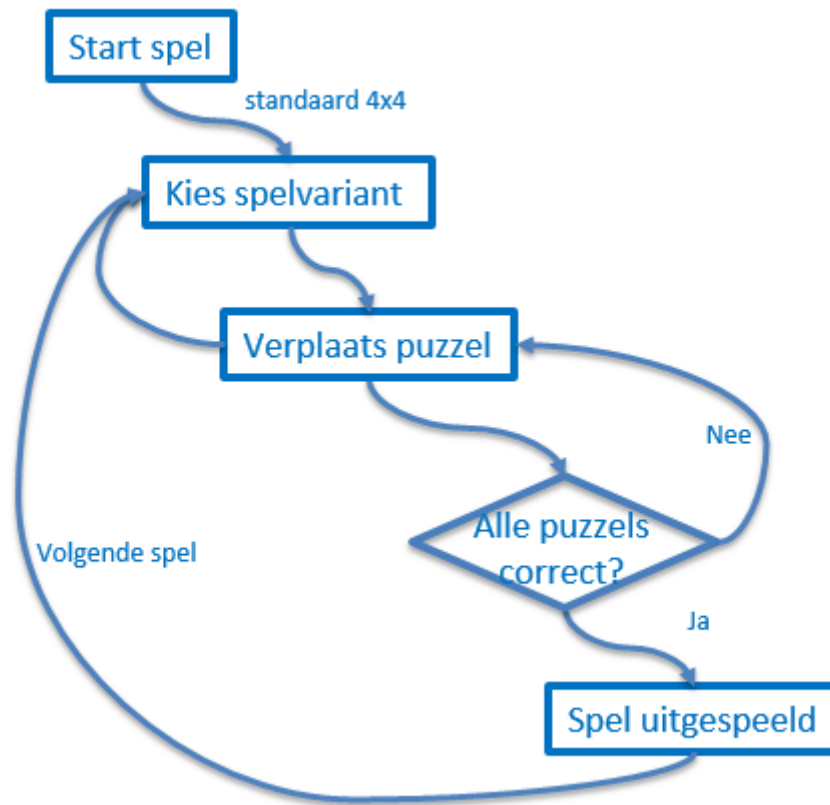
In deze versie van het spel is alleen optie 1 als extra functionaliteit ontworpen en zal dus ook gerealiseerd worden.

Meer achtergrondinformatie over deze puzzel is te vinden op

http://en.wikipedia.org/wiki/Fifteen_puzzle.

3. 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).



Figuur 1 Globale flow

Het spel kan te allen tijde beëindigd worden. Dit is daarom niet apart in de program flow opgenomen.

De bijbehorende schermontwerpen staan in het volgende hoofdstuk.

4. Schermontwerpen

Hieronder volgen de schermontwerpen op verschillende momenten tijdens het spel. Deze komen overeen met de hierboven genoemde flow.

Start spel

Zodra het spel gestart wordt zal het spel beginnen in de 4 x 4 variant en toont het bord de puzzelstukjes die reeds geshuffeld zijn.

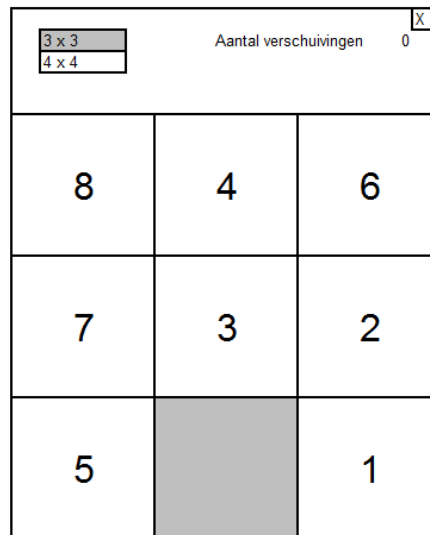
<div> <div>3 x 3</div> <div>4 x 4</div> </div>		Aantal verschuivingen 0		X
11	9	4	5	
10	1	12	6	
13	2	3		
14	8	15	7	

Figuur 2 Start spel

Kies spelvariant

De speler kan vervolgens kiezen om het spel te gaan spelen of een andere variant (3 x 3) te kiezen. Indien de speler kiest voor een andere variant zal het bord meteen een gehusseld spel laten zien in de bijbehorende variant en het aantal verschuivingen op 0 laten beginnen.

Het wisselen van spelvariant kan te allen tijde, dus ook nadat de gebruiker een aantal verschuivingen heeft gedaan.



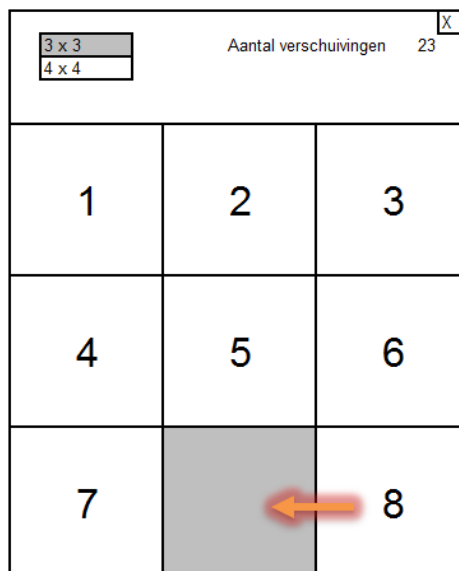
Figuur 3 Kies spelvariant

Verplaats puzzel

Het spel kan gespeeld worden door het verplaatsen van de puzzelstukjes. Een puzzelstukje wordt verplaatst als er op een puzzelstukje aangrenzend aan het open vakje geklikt wordt. In het ontwerp hierboven kunnen alleen puzzelstukje 5, 3 en 1 verplaatst worden. Na iedere verschuiving wordt gecontroleerd of het spel uitgespeeld is. Liggen nog niet alle puzzelstukjes geordend dan kan de speler doorgaan met het verschuiven van een puzzelstuk. Het aantal verschuivingen wordt na iedere verschuiving opgehoogd en verschijnt dan boven in beeld.

Spel uitgespeeld

Na enige tijd spelen wordt bijvoorbeeld onderstaande spelsituatie bereikt:



Figuur 4 Verschuif puzzelstukje

Door het puzzelstukje 8 naar links te verschuiven (zie oranje pijl) zal het spel afgelopen zijn. Het scherm zal er dan als volgt uit komen te zien:

<div> <div>3 x 3</div> <div>4 x 4</div> </div> <div> Aantal verschuivingen 24 Gefeliciteerd u heeft het spel gespeeld in 24 beurten </div> <div>X</div>		
1	2	3
4	5	6
7	8	

Figuur 5 Spel uitgespeeld

Het aantal verschuivingen is opgehoogd van 23 naar 24 en een felicitatiemelding verschijnt. Door opnieuw op een spelvariant te klikken kan een nieuw spel gestart worden.

5. Softwarebeschrijving

In dit hoofdstuk volgt een technische beschrijving van de te realiseren functionaliteit uitgewerkt in het gebruik van events, de te realiseren methoden en de te gebruiken variabelen.

Globale softwarestructuur

Het programma zal de volgende globale structuur bevatten:

- Module Events: hierin zullen de gebruikersacties worden afgevangen
- Module Teken: deze module zal alle logica bevatten om de knoppen, labels en het bord met puzzelstukken te tekenen
- Module Instellingen: hierin zullen alle scherminstellingen bijgehouden worden.
- Module Puzzelen: deze module zal de methoden bevatten die zorgen voor het verschuiven van de puzzelstukjes op de juiste plek en tevens bepalen of het spel afgelopen is. Ook de gekozen spelvariant wordt hierin bijgehouden.
- Module Algemeen: deze module zal de algemene methoden en functies bevatten die mogelijk hergebruikt kunnen worden, zoals de Hussel functie.

Events, Methoden en Variabelen

Hieronder volgt een overzicht van de onderkende events waarop het programma zal reageren en de methoden en variabelen die het gebruikt.

Module	Event/Methode/Variabele	Toelichting
Instellingen	int schermBreedte	Variabele voor het initiëren van de breedte van het scherm
	int menuHoogte	Variabele voor het initiëren van de hoogte van het menu
	int schermHoogte	Variabelen voor het bepalen van de hoogte van het scherm. Dit wordt gelijk aan de schermBreedte + menuHoogte.
Tekenen	int marge	De marge die gebruikt wordt als linker- en boven marge van de knoppen.
	int knopBreedte	De breedte van de knop, is gelijk aan 3 keer de marge
	int knopHoogte	De hoogte van de knop, is gelijk aan de marge
Puzzelen	int spelVariant	Variabele voor het bijhouden van de gekozen spelvariant (3 betekent 3x3, 4 betekent 4x4).
	int aantalVerschuivingen	Variabele voor het onthouden van het aantal puzzel verschuivingen.
	int puzzelHoogte()	Functie die de grootte van een puzzelstukje bepaalt. Deze is afhankelijk van de breedte van het scherm en de gekozen spelvariant
	void VerschuifPuzzel(int mouseX, int mouseY)	Deze methode zal de puzzel die door de speler is aangeklikt verschuiven naar de open plek op het bord
	int bepaalGeklikteObject(mouseX,mouseY)	Deze functie bepaalt of er op een spelvariant dan wel op een puzzelstukje geklikt is (PUZZEL, VARIANT3, VARIANT4).
	bool AllePuzzelsCorrect(int	Deze functie bepaalt of het spel is

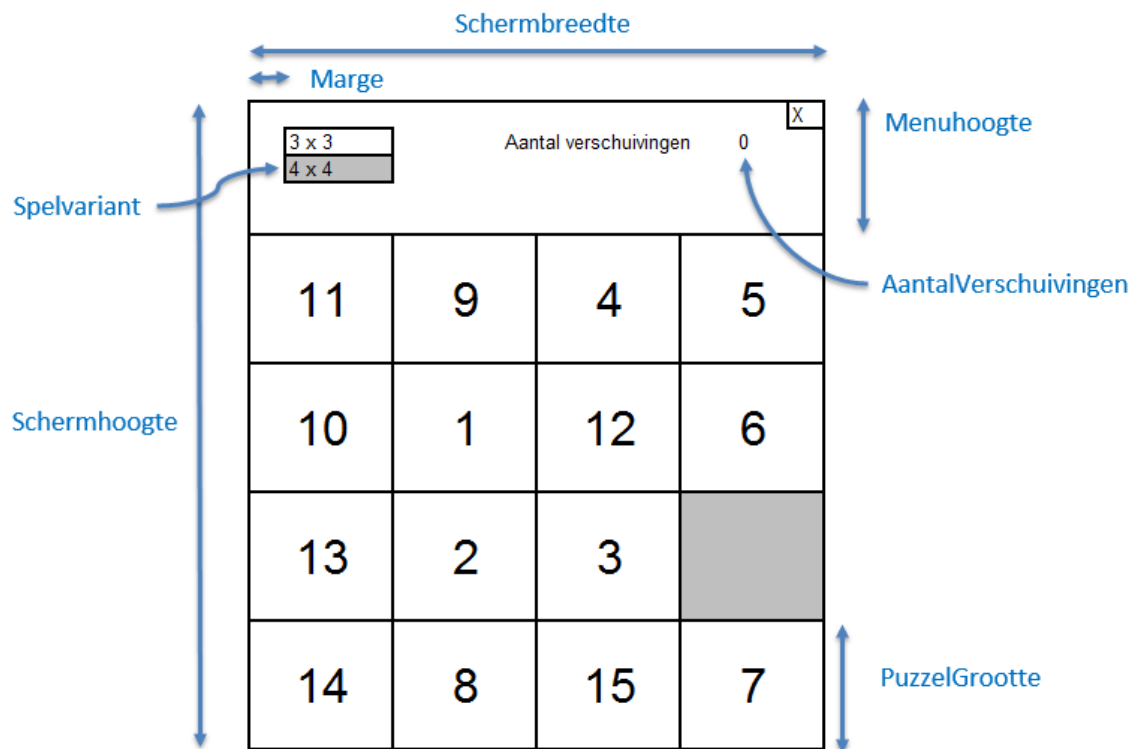
	PuzzelStukken[][])	uitgespeeld. Indien alle puzzelstukjes op de correcte plaats staan retourneert deze functie true, anders false.
Events	void Setup()	In de setup wordt de initialisatie gedaan van het spel. Hier wordt onder meer de scherm breedte en hoogte vastgesteld en defaults voor de spelvariant gezet. Vervolgens wordt de tekenmethode aangeroepen die het spel met alle opties zal tekenen
	void MouseClicked()	Als de speler met de muis klikt op het spel zal hierin vastgesteld moeten worden op welke button of puzzelstukje geklikt is. Afhankelijk van het object waarop is geklikt zal het programma ofwel een puzzelstukje verplaatsen ofwel een menu-actie uitvoeren (in geval dat er op een spelvariant is geklikt).
Tekenen	void Teken()	Deze methode tekent het gehele scherm door het aanroepen van de andere tekenmethoden.
	void TekenControls()	Deze methode zal de knoppen voor de keuze van spelvariant tekenen en de labels en tekst voor het tonen van het aantal verschuivingen.
	void TekenPuzzels(int puzzelStukken[][])	Deze methode zal op basis van een array (die de positie van de puzzelstukken representeert) de puzzelstukken tekenen.
	int PuzzelGrootte()	Deze functie retourneert de grootte (de breedte is gelijk aan de hoogte) van de puzzel. Dit is afhankelijk van de gekozen spelvariant.
	void TekenGefeliciteerd()	Deze methode geeft de melding gefeliciteerd zodra alle puzzelstukken correct staan.
Algemeen	Int [][] Hussel(int puzzelStukken[][])	Deze functie geeft op basis van een array een willekeurig gehusselde array

		terug. De hussel zal bestaan uit omwisselen van 2 willekeurige elementen uit de array. Dit omwisselen zal net zo vaak gedaan worden als de grootte van de array.
--	--	--

Samenhang schermontwerp en variabelen

Hieronder volgt een schermontwerp met daarin de variabele die benodigd zijn voor het correct tekenen van het scherm.

Deze variabelen komen overeen met de variabelen in de hierboven genoemde tabel en dienen ter verduidelijking van het ontwerp.



Programma flow

Onderstaande volgt een globaal opzet van de werking van het programma, gedeeltelijk voorzien van pseudo code:

Setup()

```
Initialisatie van schermbreedte, schermhoogte en menuhoogte.  
spelVariant = 4 // Zetten van default keuze spelvariant op 4x4  
Hussel(puzzelStukken[][])  
Teken(puzzelStukken[][])
```

MouseClicked()

```
bepaalGeklikteObject(mouseX, mouseY)  
  als VARIANT3  
    spelVariant = 3  
    puzzelStukken[][] // initieer puzzelArray van 0..8  
    puzzelStukken[][] = Hussel(puzzelStukken[][])  
  als VARIANT 4  
    spelVariant = 4  
    puzzelStukken[][] // initieer puzzelArray van 0..15  
    puzzelStukken[][] = Hussel(puzzelStukken[][])  
  als PUZZEL  
    VerschuifPuzzel(mouseX, mouseY)  
    als AllePuzzelsCorrect  
      TekenGefeliciteerd()  
  Teken()
```

Int[][] Hussel(int puzzelStukken[][])

```
Doe van 1 t/m lengte van array  
  Bepaal random puzzelX1 en puzzelY1  
  Bepaal random puzzelX2 en puzzelY2  
  Verwissel waarde;  
    puzzelStukken[puzzelX2][puzzelY2] = puzzelStukken[puzzelX1][puzzelY1]  
    puzzelStukken[puzzelX1][puzzelY1] = puzzelStukken[puzzelX2][puzzelY2]  
  (met onthouden van oude waarde)
```

Teken(int puzzelStukken[][])

```
TekenControls()  
TekenPuzzels(puzzelStukken[][])
```

TekenPuzzels(int puzzelStukken[][])

```
Teken de puzzelstukken op basis van de array en de puzzelHoogte
```

6. Testplan

Onderstaande testcases zijn ontworpen met als doel de functionaliteit van het spel te kunnen testen. Bij de uiteindelijke test wordt de kolom “werkelijk resultaat” gevuld en eventueel extra commentaar opgenomen.

TEST CASES TC 1: Spelvariant is te kiezen

FUNCTIE ONTWIKKELD DOOR

MN

TEST UITGEVOERD DOOR

<Naam>

DATUM

<Datum>

BEGINSITUATIE: Spel is opgestart

EINDSITUATIE: Spelbord wordt speelklaar gemaakt conform de gekozen speloptie

TEST	VARIABELE OF SELECTIE	WAARDE	VERWACHT RESULTAAT	WERKELIJK ¹ RESULTAAT
1.1	Speloptie kiezen	3x3	Spelbord verandert in een 3 bij 3 bord.	
			De puzzelstukken staan in willekeurig volgorde van 1 tot en met 8 genummerd	
1.2	Speloptie kiezen	4x4	Spelbord verandert in een 4 bij 4 bord.	
			De puzzelstukken staan in willekeurig volgorde van 1 tot en met 15 genummerd	
1.3	Speloptie kiezen	Gelijk aan oude speloptie	De puzzelstukken staan in een nieuwe willekeurig volgorde van 1 tot en met 15 genummerd	

¹ De kolom werkelijk resultaat zal het testresultaat bevatten bij het uitvoeren van de test.

BEGINSITUATIE: Spel is opgestart en puzzels zijn al verschoven

EINDSITUATIE: Spelbord wordt opnieuw speelklaar gemaakt conform de gekozen speloptie

TEST	VARIABELE OF SELECTIE	WAARDE	VERWACHT RESULTAAT	WERKELIJK ² RESULTAAT
1.4	Speloptie kiezen	3x3	Spelbord verandert in een 3 bij 3 bord.	
			De puzzelstukken staan in willekeurig volgorde van 1 tot en met 8 genummerd	
1.2	Speloptie kiezen	4x4	Spelbord verandert in een 4 bij 4 bord.	
			De puzzelstukken staan in willekeurig volgorde van 1 tot en met 15 genummerd	

² De kolom werkelijk resultaat zal het testresultaat bevatten bij het uitvoeren van de test.

TEST CASES TC 2: Puzzelstukje verschuiven

FUNCTIE ONTWIKKELD DOOR

MN

TEST UITGEVOERD DOOR

<Naam>

DATUM

<Datum>

BEGINSITUATIE:

Spelbord is speelklaar, open vakje bevindt zich niet aan een van de randen

EINDSITUATIE:

Puzzelstukje is verschoven naar op de open plek

TEST	VARIABELE OF SELECTIE	WAARDE	VERWACHT RESULTAAT	WERKELIJK ³ RESULTAAT
2.1	Klik op een puzzelstukje	Boven het lege vakje	Het betreffende puzzelstukje wordt naar onder verplaatst. Het lege vakje bevindt zich nu op de plek van het verplaatste puzzelstukje	
2.2	Klik op een puzzelstukje	Onder het lege vakje	Het betreffende puzzelstukje wordt naar boven verplaatst. Het lege vakje bevindt zich nu op de plek van het verplaatste puzzelstukje	
2.3	Klik op een puzzelstukje	Links van het lege vakje	Het betreffende puzzelstukje wordt naar rechts verplaatst. Het lege vakje bevindt zich nu op de plek van het verplaatste puzzelstukje	
2.4	Klik op een puzzelstukje	Rechts van het lege vakje	Het betreffende puzzelstukje wordt naar links verplaatst. Het lege vakje bevindt zich nu op de	

³ De kolom werkelijk resultaat zal het testresultaat bevatten bij het uitvoeren van de test.

			plek van het verplaatste puzzelstukje	
2.5	Klik op een puzzelstukje	Linksboven het lege vakje	Alle puzzelstukjes blijven op hun plek. Er gebeurt niets.	
2.6	Klik op een puzzelstukje	Rechtsboven het lege vakje	Alle puzzelstukjes blijven op hun plek. Er gebeurt niets.	
2.7	Klik op een puzzelstukje	Linksonder het lege vakje	Alle puzzelstukjes blijven op hun plek. Er gebeurt niets.	
2.8	Klik op een puzzelstukje	Rechtsboven het lege vakje	Alle puzzelstukjes blijven op hun plek. Er gebeurt niets.	
2.9	Klik op het lege vakje		Alle puzzelstukjes blijven op hun plek. Er gebeurt niets.	

TEST CASES TC 3: Einde spel

FUNCTIE ONTWIKKELD DOOR

MN

TEST UITGEVOERD DOOR

<Naam>

DATUM

<Datum>

BEGINSITUATIE:

Alle puzzelstukjes, op 1 na bevinden zich op de juiste plek. (Het aantal beurten is <n>.)

EINDSITUATIE:

Puzzelstukje is verschoven naar de open plek, felicitatiemelding wordt getoond

TEST	VARIABLE OF SELECTIE	WAARDE	VERWACHT RESULTAAT	WERKELIJK ⁴ RESULTAAT
3.1	Klik op het puzzelstukje	dat nog verplaatst moet worden	Het betreffende puzzelstukje wordt naar de open plek verplaatst. De open plek bevindt zich nu rechtsonder. Alle puzzelstukjes staan nu op de juiste plek. De melding "Gefeliciteerd u heeft het spel gespeeld in <n+1> beurten."	

⁴ De kolom werkelijk resultaat zal het testresultaat bevatten bij het uitvoeren van de test.

TEST CASES TC 4: Nog een keer spelen

FUNCTIE ONTWIKKELD DOOR

MN

TEST UITGEVOERD DOOR

<Naam>

DATUM

<Datum>

BEGINSITUATIE:

Alle puzzelstukjes bevinden zich op de juiste plek. Het spel is beëindigd.

EINDSITUATIE:

Spelbord wordt speelklaar gemaakt conform de gekozen speloptie

TEST	VARIABELE OF SELECTIE	WAARDE	VERWACHT RESULTAAT	WERKELIJK ⁵ RESULTAAT
4.1 (=1.1)	Speloptie kiezen	3x3	Spelbord verandert in een 3 bij 3 bord. Felicitiemelding verdwijnt. Aantal verschuivingen is teruggezet op 0.	
4.2	Klik op het puzzelstukje	Links van het lege vakje	Alle puzzelstukjes blijven op hun plek. Er gebeurt niets.	

⁵ De kolom werkelijk resultaat zal het testresultaat bevatten bij het uitvoeren van de test.