

## Slide Puzzel

---

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

Auteurs:

*Mike Trieu, Freddy de Greef,*

*Tom Peerdeman &*

*René Aparicio Saez*

Datum: 02-07-2012

## Contents

<b>1 Inleiding</b>	<b>2</b>
1.1 Wat is een Slidepuzzel? . . . . .	2
1.2 Originele concept . . . . .	2
<b>2 Opbouw applicatie</b>	<b>3</b>
2.1 Settings . . . . .	3
2.2 Highscores . . . . .	3
2.3 Play . . . . .	3
2.3.1 Game . . . . .	3
2.3.2 Tile Shuffler . . . . .	3
2.3.3 Hints . . . . .	4
2.3.4 Rotatie . . . . .	4
2.3.5 Freeze camera . . . . .	4
2.3.6 Geluid . . . . .	4
2.3.7 Win scherm . . . . .	4
<b>3 Database</b>	<b>5</b>
3.1 SQLite . . . . .	5
3.2 Opbouw tabellen . . . . .	5
3.2.1 Settings . . . . .	5
3.2.2 Highscores . . . . .	5
3.3 Voor en nadelen . . . . .	5
3.3.1 Settings . . . . .	5
3.3.2 Highscores . . . . .	5
<b>4 Solver</b>	<b>7</b>
4.1 Algoritme . . . . .	7
4.2 Manhattan distance en Linear Conflict . . . . .	7
4.3 Pad bepalen . . . . .	7
4.4 Alternatief . . . . .	7
4.5 Problemen . . . . .	7
<b>5 Vergelijking met openCV</b>	<b>8</b>
5.1 Size . . . . .	8
5.2 Difficulty . . . . .	8
5.3 Modes . . . . .	8
5.4 Shuffler . . . . .	8
5.5 Highscores . . . . .	8
<b>6 Conclusie</b>	<b>9</b>
6.1 Toevoegingen aan het originele idee . . . . .	9
6.2 Verbeteringen . . . . .	9
6.3 Conclusie . . . . .	9

## 1 Inleiding

In dit verslag wordt gekeken naar de werking en opbouw van de Slidepuzzel applicatie. Hoe het originele idee van een camerabeeld omzetten tot een slidepuzzel is uitgebreid naar allerlei verschillende modes, moeilijkheidsgraden en afmetingen en hoe deze puzzels op te lossen. Daarnaast wordt vergeleken hoe deze applicatie werkt ten opzichte van de openCV versie van de 15-puzzle (een vier bij vier slidepuzzel) voor Android.

### 1.1 Wat is een Slidepuzzel?

Een slidepuzzel, of schuifpuzzel, is in feite een plaatje dat in even grote blokjes, of tiles, is verdeeld. Eén van deze tiles wordt uit het originele plaatje verwijderd, waardoor er met de tiles geschoven kan worden. De puzzel bestaat uit  $N$  bij  $M$  tiles, afhankelijk van het originele plaatje en de grootte van ieder blokje. De meest voorkomende slidepuzzels zijn de 8-puzzel en de 15-puzzel. Deze bestaan respectievelijk uit drie bij drie en vier bij vier tiles.

### 1.2 Originele concept

Het originele idee voor deze slidepuzzel Android applicatie was het statische plaatje te vervangen door een dynamisch veranderend camerabeeld. Hierbij zouden enkele verschillende moeilijkheidsgraden komen. Een mogelijkheid om met hints te spelen, de normale versie (met camerabeeld) en een moeilijke versie met roterend beeld. Daarnaast zou het alleen mogelijk zijn om te spelen met een veld van drie bij drie of vier bij vier tiles. Dit in verband met de rotaties, het is immers overzichtelijker om een vierkant beeld te roteren dan om een rechthoekig beeld te roteren. Grottere speelvelden werden al uitgesloten, omdat de tiles dan te klein zouden worden voor sommige Android devices. Hierbij zouden voor het speleffect enkele geluidseffecten worden toegevoegd.

## 2 Opbouw applicatie

Wanneer de applicatie gestart is, wordt er een menu gedisplayed. De opties in het main menu zijn: play, settings, highscores, how to play en exit.

### 2.1 Settings

In het settings menu kan de moeilijkheidsgraad, game formaat en game modus worden aangepast. Voor de moeilijkheidsgraad zijn er drie niveaus geïmplementeerd:

- De easy moeilijkheidsgraad laat hints toe
- De normal moeilijkheidsgraad laat geen hints toe
- De hard moeilijkheidsgraad draait het beeld om de tien seconden 90 graden

N.B. hints zijn ook niet toegestaan in hard

Het formaat van de game kan aangepast worden naar een drie bij drie of een vier bij vier schuifpuzzelveld. Hierbij kan gekozen worden uit twee verschillende gamemodes. De live image mode en de gallery image mode. De live image mode maakt gebruik van het inkomende camerabeeld. De puzzel zal live meeveranderen tijdens het spelen. Bij de gallery image mode kan een foto uit het geheugen van de Android device worden ingeladen als puzzel. De settings worden opgeslagen in een database (3). Dit zorgt ervoor dat de settings die door de gebruiker zijn ingesteld worden onthouden, zelfs als de applicatie wordt afgesloten.

### 2.2 Highscores

Highscores worden bepaald door de tijd die nodig is om een puzzel op te lossen. De highscores van de applicatie zijn opgedeeld in verschillende categorieën. Ten eerste wordt dit bepaald door de moeilijkheidsgraad. Per moeilijkheidsgraad is vervolgens weer een onderverdeling tussen het drie bij drie en het vier bij vier speelveld. Per categorie worden maximaal de vijf beste tijden getoond, met de snelste tijd bovenaan. Hierdoor kan de gebruiker zien wat de beste tijden zijn in een specifieke moeilijkheidsgraad en game formaat.

### 2.3 Play

De play knop brengt de gebruiker naar de game zelf. Hieronder worden de features van de game beschreven en uitgelegd.

#### 2.3.1 Game

De schuifpuzzel zelf gebruikt informatie van de settings om het speelveld te creëren. Het beeld wordt verdeeld in tiles. Het aantal tiles wordt bepaald door het gekozen spel formaat. Uit het originele beeld wordt de tile die zich rechts onder bevindt zwart gemaakt. Vervolgens wordt het beeld gehusseld.

#### 2.3.2 Tile Shuffler

Als de tiles in een vier bij vier veld random worden neergelegd is het mogelijk dat daar een niet oplosbare puzzel uit voortkomt. Daarom is besloten om zowel voor de drie bij drie als de vier bij vier dezelfde manier van husselen te gebruiken, die het onmogelijk maakt om een onoplosbare puzzel te creëren. De puzzel wordt vanuit zijn beginstaat, waar alle tiles op de goede locatie liggen, willekeurig verschoven, mits het mogelijk is om te schuiven. Nadat er gehusseld is wordt gecontroleerd of er niet toevallig het eindresultaat uit is voortgekomen, is dit wel het geval dan wordt overnieuw gehusseld. OpenCV [1] kent een slidepuzzel van vier bij vier tiles die gebruik maakt van een ander algoritme om de puzzel te husselen [2]. Hierbij kunnen de tiles willekeurig op het veld worden geplaatst waarna gekeken wordt of de puzzel op te lossen valt.

### 2.3.3 Hints

In easy mode is er een hint systeem ingebouwd (4). Bij het gebruik van deze knop in wordt de best mogelijke zet getoond op dat moment. Herhaling zal uiteindelijk de puzzel oplossen.

### 2.3.4 Rotatie

Als er is gekozen om te spelen in hard, dan wordt het originele beeld iedere tien seconden 90 graden gedraaid. Dit zorgt ervoor dat de gebruiker zich opnieuw moet oriënteren. Dit geeft een compleet nieuwe draai aan het schuifpuzzel concept en maakt de schuifpuzzel aanzienlijk moeilijker.

### 2.3.5 Freeze camera

In de live image mode van de schuifpuzzel is er de mogelijkheid om de freeze button te gebruiken. Hiermee kan het beeld worden vastgezet zodat als het ware met zelfgemaakte foto gespeeld wordt. Het gebruik van deze knop is bedacht om het spelersgemak te vergroten. Immers is het mogelijk dat het beeld continu veel heen en weer beweegt, bijvoorbeeld tijdens het reizen. Dit zorgt er echter voor dat er geen onderscheid meer is tussen de gallery image en live image mode. Mede hierdoor is er gekozen om in de highscores geen onderscheid te maken tussen live image en gallery image puzzles.

### 2.3.6 Geluid

Als tijdens het spelen het geluid aan staat op de Android device, zijn er enkele geluiden te horen. Elke keer als de gebruiker een tile in beweging zet, wordt er een schuifgeluid afgespeeld. Daarnaast wordt er in de hard mode bij elke rotatie een geluidsfragment afgespeeld. Zodra de puzzel is opgelost wordt er een geluidsfragment afgespeeld om aan te geven dat het spel voorbij is.

### 2.3.7 Win scherm

Zodra de puzzel is opgelost wordt er een scherm weergegeven die de gebruiker laat zien hoe lang er over is gedaan om de puzzel op te lossen. De gebruiker kan zijn naam invullen. Mocht de tijd één van de snelste tijden zijn dan wordt deze tijd met bijbehorende ingevoerde naam getoond in het highscores menu (2.2). Vervolgens kan er gekozen worden terug te keren naar het main menu door middel van de knop 'back', of om een nieuw spel te starten door middel van de knop 'new game'. In beide gevallen wordt de tijd met bijbehorende naam goed opgeslagen. Als er wordt gekozen om een nieuw spel te starten, dan begint direct een nieuw spel met dezelfde settings als het afgelopen spel.

## 3 Database

### 3.1 SQLite

Om de settings en de highscores bij te houden gebruikt het programma een database. De database die hiervoor gebruikt wordt is de SQLite database. De SQLite database is een lichtgewicht ingebouwde database in Android. De SQLite database komt in grote lijnen overeen met andere databases, er worden query's in SQL (Structured Query Language) gestuurd naar de database. Deze database zal dan deze query's afhandelen, zo nodig zal bij een SELECT query het resultaat teruggegeven worden. Het uitvoeren van operaties op deze database is in Android niet heel lastig. Als de database aangemaakt is en er is een actieve connectie, kunnen er via toegangsmethoden operaties uitgevoerd worden. De operaties die het programma gebruikt en die gedefinieerd zijn in de gebruikte klasse zijn insert, update en select. Er is dus voor deze operaties in principe geen kennis van SQL nodig. Om de tabellen in de database aan te maken is wel degelijk kennis van SQL nodig. De SQLite klassen maken alleen de database zelf aan, om tabellen aan te maken moet een handmatig gemaakte CREATE TABLE query uitgevoerd worden.

### 3.2 Opbouw tabellen

#### 3.2.1 Settings

Voor de settings wordt gebruik gemaakt van een tabel met slechts één entry. Deze entry representeert de huidige settings. De tabel bestaat dan ook uit drie kolommen, namelijk difficulty, size en mode. De difficulty bestaat uit een string van minimaal vier tekens en kan de waardes EASY, NORMAL en HARD bevatten. De size kolom bestaat uit een integer, deze kan de waardes 3 en 4 bevatten. Hierbij stelt de 3 een drie bij drie veld voor en de 4 een vier bij vier veld. De mode kolom bevat de game mode, deze bestaat uit een integer met de waarde 0 wat staat voor live beeld, of 1 wat staat voor een plaatje uit de gallery.

#### 3.2.2 Highscores

De highscores tabel bevat ongeveer alle velden van de settings tabel, dit omdat de highscores geselecteerd moeten worden op spel type. De kolommen die nog extra staan in de highscores tabel zijn name en time. De name kolom bevat simpelweg de naam die de speler heeft ingevoerd bij het winnen van het spel. De time kolom bevat de tijd in seconden die de speler heeft gedaan over het oplossen van de puzzel. In tegenstelling tot de settings tabel kan de highscores tabel meer dan één entry bevatten. Omdat dezelfde gebruiker meerdere entry's in de highscore kan hebben en dus ook dezelfde naam kan gebruiken, moet er een andere manier dan naam zijn om de entry's uit elkaar te houden. Hiervoor is de extra kolom id gebruikt. De id kolom is een kolom die automatisch opgehoogd wordt bij het invoegen van een highscore entry, zo kunnen er nooit 2 compleet identieke rijen staan in de database.

### 3.3 Voor en nadelen

#### 3.3.1 Settings

Aangezien de settings tabel maar één entry bevat is het redelijk makkelijk om de settings in te laden, er hoeft simpelweg maar één rij geselecteerd te worden. Het nadeel van de op database gebaseerde settings is dat bij elke wijziging van de settings een update van de database uitgevoerd moet worden. Dit nadeel zou je kunnen opheffen door de update alleen plaats te laten vinden bij het afsluiten van de app. Dit zou het spel zelf niet schaden aangezien de actuele settings toch in het geheugen worden opgeslagen, het nadeel van deze methode is alleen dat bij een crash (bijvoorbeeld lege batterij) het niet zeker is of de update is gedaan.

#### 3.3.2 Highscores

Het gebruik van de database voor de highscores is daarentegen wel goed verdedigbaar. Het invoegen van de highscores is simpel door de toegangsmethoden van de SQLite database. Het is

ook simpel om de juiste entry's te selecteren doordat er in de query's een where conditie kan worden meegegeven. Het lastige zoekwerk wordt dan afgehandeld door de database zelf. Het enige nadeel op dit moment is dat er geen verwijder methode in de app is voor de highscore entry's. Dit zou kunnen leiden tot een enorme database waardoor het laten zien van de highscores langzaam zou gaan. Verder is er op mobiele apparaten geen onbeperkte opslagruimte, dus zou de database uiteindelijk niet meer kunnen groeien. Dit is niet erg gebruiksvriendelijk.

## 4 Solver

### 4.1 Algoritme

Het algoritme wat wordt gebruikt om de puzzel op te lossen is het Iterative Deepening A\* search algoritme. Het algoritme werkt als volgt: stel dat elke staat van de puzzel een knoop is, dan zal de staat waaruit op hint wordt gedrukt de beginknoop zijn. In dit algoritme wordt vanuit de begin knoop de boom doorlopen zover als de dieptegrens is ingesteld. Vanaf de huidige knoop wordt de volgende knoop bezocht als deze voldoet aan een bepaalde Manhattan distance waarde. Dit blijft zich herhalen tot de eindknoop (de oplossing) of de dieptegrens is bereikt. Als de dieptegrens is bereikt en de eindknoop is niet gevonden dan wordt er opnieuw gezocht met een hoger geaccepteerde Manhattan distance waarde. Meer knopen zullen dan bezocht worden. Deze stappen herhalend wordt de hele boom doorlopen. Mits de oplossing niet gevonden is en de hele boom doorlopen is, wordt de dieptegrens 2 stappen groter. Zo gauw als de oplossing gevonden is, stopt het algoritme. Voor het oplossen van de puzzel wordt een kopie van het speelveld gebruikt als begin knoop. Er wordt dan gezocht door middel van deze kopie, zodat de gebruiker zijn puzzel niet kwijt raakt.

### 4.2 Manhattan distance en Linear Conflict

De Manhattan distance wordt berekend door van een tile te bekijken hoeveel zetten hij nog nodig heeft om naar de oplossingspositie te komen als er geen andere tiles op het bord staan. Dit wordt voor elke tile berekend en bij elkaar opgeteld. Het resulterende getal is de Manhattan distance. Een toevoeging hierop is Linear Conflict. Bij Linear Conflict wordt gekeken of twee tiles, die al op de goede lijn staan maar niet op de juiste plaats, op dezelfde lijn staan. De lijn kan een rij of een kolom zijn. Stel dat de eerste rij gevuld is met tile 3 met rechts daarvan tile 2. Linear Conflict zou dan tenminste 2 zetten tellen bij de Manhattan distance voor deze twee conflicterende tegels. De heuristiek die gebruikt wordt voor het algoritme is deze Manhattan distance waarde. Deze distance helpt het algoritme met het sneller vinden van de beste oplossing.

### 4.3 Pad bepalen

Als de eindstatus is gevonden dan is er een pad bekend. Elke beweging op dit pad wordt opgeslagen in het path array. Hierin staan alle stappen die genomen moeten worden om bij de oplossing te komen. Het eerste element wordt gegeven als de beste zet op dat moment.

### 4.4 Alternatief

Het algoritme wat nu gebruikt wordt om de puzzel op te lossen is nogal traag. Een verbetering zou zijn om het A\* (A-ster) algoritme te gebruiken. Dit algoritme vindt de beste oplossing. In de gebruikte code kon het A\* algoritme niet gebruikt worden bij een vier bij vier schuifpuzzel. Een oplossing hiervoor zou zijn om met een algoritme van de vier bij vier puzzel een drie bij drie puzzel te maken door de eerste rij en de eerste kolom te vullen. Deze puzzel kan dan worden opgelost met behulp van het A\* algoritme.

### 4.5 Problemen

Zoals eerder aangegeven, is het Iterative Deepening A\* search algoritme vrij traag. Dit resulteert erin dat een vier bij vier puzzel veel tijd nodig heeft om te worden opgelost. Dit kost zoveel tijd dat er is besloten om voor het vier bij vier veld geen hints toe te laten. Uiteindelijk werkte het hele algoritme niet, het kan zo zijn dat als er op hint wordt gedrukt in de drie bij drie easy modus, maar de puzzel zal nooit opgelost worden.

## 5 Vergelijking met openCV

OpenCV biedt vrij te downloaden applicaties aan voor onder andere Android. Zo ook een applicatie die met een live videobeeld een slidepuzzel maakt. Er zijn echter vele verschillen tussen de applicatie van openCV en onze applicatie.

### 5.1 Size

OpenCV levert een 15-puzzle (vier bij vier schuifpuzzel). Ze staan dus alleen een vier bij vier veld toe, waar het bij onze applicatie mogelijk is om ook een drie bij drie veld te kiezen. De tiles die openCV gebruikt zijn daarnaast ook geen vierkantjes maar rechthoeken, bij onze applicatie zijn dit wel vierkantjes.

### 5.2 Difficulty

De 15-puzzle van openCV kent slechts één enkele moeilijkheidsgraad. Deze valt te vergelijken met onze normale modus. OpenCV biedt daarnaast de mogelijkheid aan om getallen weer te geven op de tiles die aangeven waar ze terecht moeten komen. Onze versie bevat meerdere moeilijkheidsgraden. De easy mode biedt de mogelijkheid aan de gebruiker om de best mogelijke zet op dat moment aan te geven. De hard mode maakt het de gebruiker moeilijker door iedere tien seconden een rotatie uit te voeren op het beeld.

### 5.3 Modes

De live 15-puzzle van openCV biedt geen mogelijkheid om een foto in te laden om mee te spelen. De applicatie is immers puur gemaakt om met live beeld te spelen. Onze applicatie heeft uiteindelijk ook een mogelijkheid gekregen om met een foto op de Android device te spelen.

### 5.4 Shuffler

Het creëren van een slidepuzzel gebeurt door het willekeurig plaatsen of verschuiven van tiles. OpenCV plaatst alle tiles op willekeurige locaties in het speelveld. Vervolgens wordt er gechecked of het gemaakte veld op te lossen valt. Hiervoor wordt een wiskundige berekening gebruikt [2]. Valt het bord op te lossen dan begint het spel, anders wordt het willekeurig plaatsen van tiles herhaald tot er wel een speelbaar veld uit komt. Onze applicatie maakt gebruik van een andere methode om een speelveld te maken. Er wordt uitgegaan van het originele beeld. Vervolgens wordt willekeurig een richting gekozen om naartoe te schuiven. Is het mogelijk deze richting op te schuiven dan gebeurd dit. Is dit niet mogelijk dan wordt opnieuw een willekeurige richting gekozen. Dit proces herhaaldt zich totdat er 100 willekeurige stappen zijn gedaan. Vervolgens wordt bekeken of de puzzel niet bij toeval het originele beeld oplevert. Als dit wel het geval is dan wordt er overnieuw gehusseld.

### 5.5 Highscores

De applicatie van openCV houdt niet bij hoelang een gebruiker erover doet om de puzzel op te lossen. Dit resulteert erin dat er geen highscores bij worden gehouden. Onze applicatie houdt wel een tijd en highscores bij van de gebruiker. Dit levert voor de gebruiker een leukere speelervaring op, hij zal immers steeds sneller en beter willen presteren.

## 6 Conclusie

### 6.1 Toevoegingen aan het originele idee

Al met al bevat de applicatie alle onderdelen die in het originele idee naar voren waren gekomen. Daarnaast zijn er ook nog enkele toevoegingen gemaakt die de applicatie verbeteren. De grootste wijziging is het toevoegen en bijhouden van highscores door middel van de tijd die nodig is om een puzzel op te lossen. Daarnaast is de freeze functie tijdens het spelen ook een handige toevoeging voor de gebruiker. Tegelijkertijd met het toevoegen van de freeze functie kwam het idee om ook de mogelijkheid te bieden om met foto's uit het geheugen van de Android device te spelen. Hiermee krijgt de gebruiker ook de mogelijkheid om de slidepuzzel op de klassieke manier te spelen.

### 6.2 Verbeteringen

Er zijn nog enkele verbeteringen mogelijk bij deze applicatie. Zo groeit de database momenteel net zo lang door totdat er geen geheugen meer vrij is. Er bevindt zich ook een fout in de gallery image mode. Als er voor deze mode is gekozen en de Android device wordt in rechtopstaande positie gehouden tijdens het kiezen van een plaatje uit het geheugen, dan komt het gekozen plaatje na het kiezen niet in beeld. Er wordt dan een camerabeeld zichtbaar dat niet is opgedeeld in tiles en niet werkt als een puzzel. Ditzelfde gebeurt als de Android device niet is ingesteld om het scherm automatisch te laten draaien.

### 6.3 Conclusie

De applicatie voldoet aan de verwachtingen. Er kan op een eenvoudige en leuke manier met live of met een vast beeld gespeeld worden. In vergelijking met openCV kan deze applicatie beduidend meer. De vergelijking met openCV is in sommige opzichten echter niet terecht omdat er bij openCV slechts een eenvoudig voorbeeld van een vier bij vier slidepuzzel wordt gegeven.

## References

- [1] opencv 15-puzzle sample. <https://code.ros.org/trac/opencv/browser/trunk/opencv/samples/android/15-puzzle?rev=6088>.
- [2] Mark Ryan. Solvability of the tiles game. <http://www.cs.bham.ac.uk/~mdr/teaching/modules04/java2/TilesSolvability.html>.