

Binaire Brainiac

Probleemstelling

- Het maken van een algoritme om Binairo's op te lossen
- Er wordt gewerkt met de officiële Binairo regels
- Het startveld wordt handmatig ingevoerd en kan elke maat aannemen zolang de lengte en breedte gelijk zijn

Regels Binairo

1. Elke cel moet een 0 of een 1 bevatten
2. Er mogen niet meer dan twee dezelfde cijfers aan elkaar grenzen op één lijn
3. Elke rij en elke kolom moet evenveel nullen als enen bevatten
4. Elke rij is uniek en elke kolom is uniek. De invulling van een rij en kolom kunnen wel overeenkomen
5. De puzzel moet opgelost kunnen worden zonder te gokken

Algoritme - algemeen

- Loopt om en om alle regels af, op de eerste regel na
 - De tweede regel is het belangrijkste
- Om verticaal en horizontaal te controleren wordt er in de code een gedraaide versie van het veld bijgehouden, om te controleren of de kolommen ook aan de eisen voldoen
- Wanneer er in een van de velden (oorspronkelijk of gedraaid) een regel wordt toegepast en een veld wordt ingevuld moet dit ook gebeuren in het andere veld
 - Er worden dus constant 2 versies van het speelveld gecontroleerd en geüpdatet
- De functie RuleCheck() blijft over alle regels lopen tot er geen punten (lege velden) meer aanwezig zijn en de puzzel is opgelost

Algoritme - regels

- Regel 2 (afbeelding A) gaat elk karakter in het veld los af en controleert of dit een punt of een cijfer is
 - Bij een cijfer wordt er gezocht naar aangrenzende identieke cijfers, indien dit het geval is wordt het andere cijfer hier omheen ingevuld. Er wordt hierbij rekening gehouden met de dimensies van het speelveld, zodat er niet buiten het veld wordt gecontroleerd of geschreven, om errors en schrijffouten te voorkomen. Dit is hoe duo's worden gevonden
 - Bij een punt zoekt het algoritme naar omliggende identieke cijfers en als dit het geval is, wordt het andere cijfer geschreven. Zo voorkomt het dat er 3 van hetzelfde cijfer op een rij komen te liggen
- Regel 3 (afbeelding B) vult rijen en kolommen aan wanneer het maximale aantal van een cijfer daar al gebruikt is. Dit kan dus zijn bij een rij of kolom waar alle vlakken op één na zijn ingevuld, maar ook al bij een rij of kolom waar de helft is gevuld met een cijfer
- Regel 4 controleert of alle rijen en kolommen identiek zijn onderling, zoals beschreven in spelregel 4. Wanneer dit het geval is rondt de functie succesvol af

Gebruik Applicatie

- Simpel, niet erg gestroomlijnd voornamelijk door het handmatige invoeren van het speelveld bij het opzetten ervan
- De functie FieldGeneration() is wat het veld genereerd door een lijst met daarin lijsten te maken met de 1 karakter strings “.”, “1” of “0”. Na het invullen van het speelveld wordt er tevens een overzicht van geprint, en korte tijd hierna de oplossing
- Wanneer de puzzel niet op te lossen is volgens de standaard regels zal de tekst “Solving...” blijven verschijnen, waarna het programma moet worden onderbroken met Ctrl + C

	1				
0				1	1
0	0			1	
		1			0
			1		
1				0	

1	1	0	1	0	0
0	1	0	0	1	1
0	0	1	0	1	1
1	0	1	1	0	0
0	1	0	1	1	0
1	0	1	0	0	1

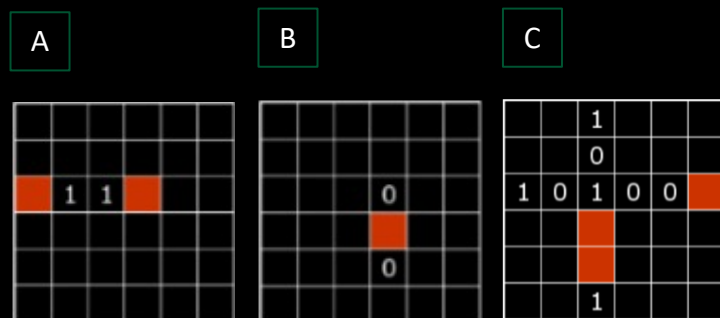
6 x 6 Binairo vóór en ná het oplossen

Opzet

- Opgeven van dimensies
- Rij voor rij invullen van het veld
 - Lege velden worden aangegeven met een punt
 - Alle tekens worden in de code geregistreerd als strings

Dit ziet eruit als volgt:

```
[ '.', '1', '.', '.', '.', '.', '.']  
[ '0', '.', '.', '.', '1', '1']  
[ '0', '0', '.', '.', '1', '.']  
[ '.', '.', '1', '.', '.', '0']  
[ '.', '.', '.', '1', '.', '.']  
[ '1', '.', '.', '.', '0', '.']
```



<https://www.binairepuzzel.net/tips.php>

```
Solving...  
[ '1', '1', '0', '1', '0', '0']  
[ '0', '1', '0', '0', '1', '1']  
[ '0', '0', '1', '0', '1', '1']  
[ '1', '0', '1', '1', '0', '0']  
[ '0', '1', '0', '1', '1', '0']  
[ '1', '0', '1', '0', '0', '1']
```

De opgeloste puzzel als output