

Plan van aanpak

Probleem

Stel je voor je bent een groot fan van Sudoku en andere puzzels en je gaat dus door boeken heen alsof het niks is. Dit is het geval voor mijn oma ze is super snel met het oplossen en heeft dus erg veel afgeronden puzzelboeken thuis liggen die niet meer gebruikt worden. Dit kan best een grote kostenpost worden maar ook worden de sudoku boeken steeds minder uitdagend. Het zou dus voordelig en een leuke uitdaging zijn als er een programma gemaakt word die unieke sudoku borden kan genereren en deze ook kan oplossen als ze te moeilijk zijn.

Algoritme

Na verdieping in het oplossen van sudoku's heb ik een wetenschappelijk paper gevonden waarin een algoritme omschreven word genaamd: Recursive Backtracking for Sudoku. Dit is een vorm van een brute force algoritme die getallen probeert en als het faalt stappen terug neemt en andere getallen probeert. Hierdoor werkt het programma naar een antwoord toe. Als dit algoritme goed geïmplementeerd is kost het gemiddeld 20.365s om de puzzel op te lossen.

De paper waarom ik dit baseer staat als bron onderaan genoteerd.

**** OUD ****

Mijn algoritme is een combinatie van rule based en hypothesis (Brute Force). Dit doe ik omdat hypothesis uit zichzelf best inefficient is maar als we een paar basis rules toevoegen die mensen ook gebruiken om het op te lossen word dit al meteen een stuk sneller. Dit zijn basis regels zoals, binnen een bepaalde subgrid mag niet 2 keer hetzelfde nummer of binnen dezelfde row/column mag niet hetzelfde nummer. Geen invalid nummers (niet boven de 9 en onder de 1).

**** OUD ****

Taken

Er moeten een aantal grote taken gemaakt worden en een aantal wat kleinere. Ik ga dit opdelen door middel van de MoSCoW planning methode. Hierin ga ik alle taken afbakenen zodat er geen onduidelijkheden komen over wat er wel en niet gemaakt gaat worden.

Taak	MoSCoW	Tijdinschatting
Het Recursive Backtracking algoritme	Must	32 uur. (4 werkdagen)
Het rule based algoritme uitbreiding	Could	20 uur.
UI voor het programma	Must	32 uur. (4 werkdagen)
Goede UX design	Should	5 uur.
Mogelijkheid voor Hints	Could	10 uur.

Taak	MoSCoW	Tijdinschatting
Animaties voor het algoritme	Will not	12 uur.
Visualisatie werking algoritme	Could	6 uur.
Drag & Drop	Will not	6 uur.
In enkel cell meerdere in geheugen zetten	Could	6 uur.

Planning

WEEK 0

In de voorbereiding week hoop ik al wat stappen gezet te hebben richting het realiseren van de basis van het sudoku programma. Zoals het opzetten van de GitHub en opdelen van de Classes.

WEEK 1

In de eerste week wil ik me gaan focussen op het maken van een ui. Deze wil ik afhebben zodat ik het algoritme meteen kan implementeren en testen. Ook wil ik me in de eerste week proberen uit te dagen met wat COULDS vanuit de takenlijst.

WEEK 2

In de tweede week wil ik me gaan focussen op het realiseren van het algoritme en de uitbreiding van het algoritme. Ook als ik tijd over heb wil ik me gaan focussen op het kunnen vragen om hints.

WEEK 3

In de laatste week wil ik gaan werken aan het afronden van het project. Hieronder valt alles vanaf bug fixes naar het maken van een mooie poster en laatste documentatie afronden.

Risico's

Een van de grootste risico's die voor mij van toepassing zijn is dat ik vaak super veel energie kan verspillen in het verder werken aan optionele taken omdat die super interessant lijken. Dit kan een grote impact hebben omdat dit ervoor kan zorgen dat ik de basis niet afkrijg. Dit ga ik deze keer opvangen door de taken goed op te delen en te verspreiden over de weken zodat ik ze altijd nog op zou kunnen pakken maar niet direct op mijn werkpad komen.

Nog een risico is dat het algoritme niet zo effectief blijkt te zijn. Dit heeft een medium impact want het betekent dat het algoritme wel nog steeds effectief is maar gewoon niet zo presteert als gewenst. Wat ik dan ga doen is op basis van de tijd het algoritme verder onderzoeken en waar nodig wat uitbreiden met slimmer rules en de code optimaliseren waar nodig.

Bronnen

<http://byteauthor.com/2010/08/sudoku-solver/>

<https://www.geeksforgeeks.org/ml-understanding-hypothesis/>

[#~:text=A%20hypothesis%20is%20a%20function,have%20imposed%20on%20the%20data.](#)

<https://www.productplan.com/glossary/moscow-prioritization/>

[#~:text=MoSCoW%20stands%20for%20four%20different,will%20not%20have%20right%20now.%E2%80%9D](#)

Het artikel

<http://www.journal.bonfring.org/abstract.php?id=2&archiveid=484>

