

MAIZE

Rudolf Sijbrandij

Probleembeschrijving

Mijn nichtje is helemaal fan van doolhoven maken en het oplossen daarvan. Mijn oom en tante vragen mij regelmatig om op haar te passen en dan vraagt ze mij telkens om doolhoven voor haar te maken om op te lossen, alleen ben ik daar niet heel goed in en maak ze altijd te makkelijk en natuurlijk doe ik er dan veel te lang over terwijl zij ze dan binnen 20 seconden oplost... daar moet ik dus een oplossing voor verzinnen.

Het probleem is dat ik geen zin meer heb in het maken van doolhoven voor mijn nichtje omdat het veel tijd kost en zij ze altijd te makkelijk vind, dus ik moet een programma maken dat het maken van het doolhof voor mij doet, met nog wat extra toevoegingen.

Eisen

- Zelf instelbare afmetingen
- Steeds een ander doolhof
- Niet te makkelijk
- Iets om snel de uitweg te vinden
- Ziet er mooi uit
- Het zelf makkelijk kunnen oplossen (zelf een character door het doolhof heen laten bewegen)

Algoritme

Ik heb voor het doolhof maak algoritme het Depth First Search algoritme gebruikt, tevens voor het oplos algoritme heb ik die ook gebruikt. Dat algoritme werkt als volgt:

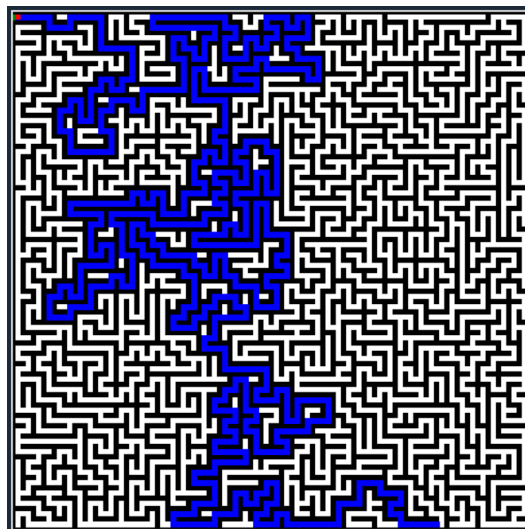
1. Je kiest de eerste cel, wat in mijn geval altijd [0, 0] is, dus linksboven, je markeert deze cel als bezocht en stopt hem in een stack.
2. Zolang de stack niet leeg is doe het volgende;
 - a. Pop een cel uit de stack en maak de cel de huidige cel
 - b. Zolang de huidige cel burens heeft die niet bezocht zijn;
 - i. Stop de huidige cel in de stack
 - ii. Kies 1 van de onbezochte burens
 - iii. Maak die buurcel de huidige cel
 - iv. Markeer de nieuwe huidige cel als bezocht en stop hem in de stack

Oftewel je blijft een route volgen en zodra je niet meer verder kan ga je terug en markeer je het hele deel waar je terug gaat als bezocht en ga je daar niet meer langs.

Voor het maze generation algoritme werkt het iets anders dan het solve algoritme; bij het generation algoritme onthoud je elke cel waar je langs bent gekomen en vul je dat direct in als een geldig pad van het doolhof, bij het solve algoritme beloop je dat pad en als je terug moet vergeet je dat dat deel was van datgene wat je moet teruggeven als oplossing.

Resultaat

Alle eisen van de opdrachtgever zijn voldaan, want je kan aan het begin van het programma de afmetingen van het doolhof zelf instellen, ook is het doolhof steeds willekeurig dus telkens een ander doolhof. Aangezien je de afmetingen zelf kan veranderen kan je het zo moeilijk maken als je wil, ook is de "Disjointed" optie aanwezig, daarmee kan je een disjointed doolhof laten genereren. Om snel de uitweg te vinden kan je op "Solve" drukken en dan maakt het programma een weg van het begin tot eind. De UI ziet er gewoon clean uit dus beter word dat niet echt. Verder is er ook een character dat je zelf kan besturen met de pijltjes toetsen op je toetsenbord om het doolhof zelf door te lopen.



Bronnen

Cope, G. (z.d.). MAZE GENERATION ALGORITHM - DEPTH FIRST SEARCH. algosome.
<https://www.algosome.com/articles/maze-generation-depth-first.html>

Kozlova, A., & Brown, J. A. (2015, augustus). Examination of Representational Expression in Maze Generation Algorithms. Researchgate. Geraadpleegd op 14 juni 2022, van
https://www.researchgate.net/publication/281101395_Examination_of_Representational_Expression_in_Maze_Generation_Algorithms

Gabrovšek, P. (2017). Analysis of Maze Generating Algorithms.
<http://ipsitransactions.org/journals/papers/tir/2019jan/p5.pdf>