

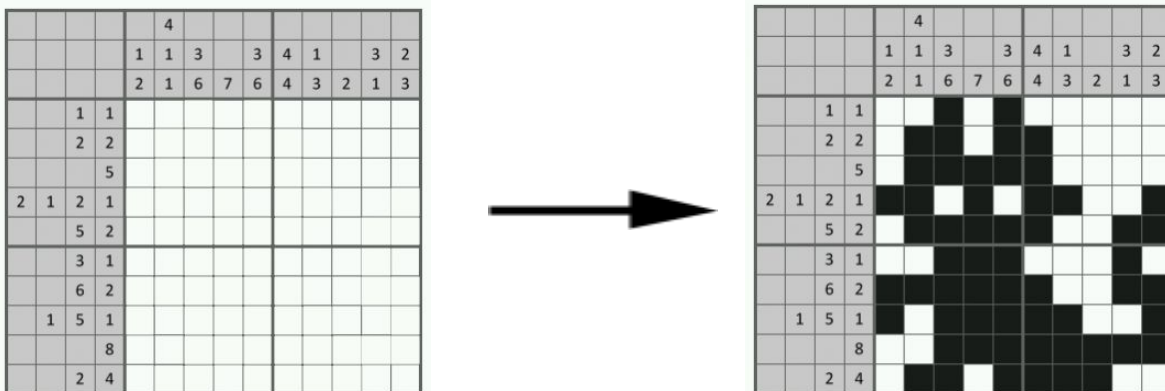
Programmeerproject Java

Academiejaar: 2019-2020

Projecttitel: Nonogram

Projectcode: NON

SPELBESCHRIJVING



Een nonogram is een japanse beeldpuzzel waarbij het de bedoeling is een leeg rooster opgevuld te krijgen met 'pixels' aan de hand van de getallen die links en boven het rooster staan. Deze getallen geven aan hoeveel (groepen van) pixels er in de rijen en kolommen mogen staan. Het voorbeeld hierboven maakt veel duidelijk alsook de uitleg op wikipedia: <https://nl.wikipedia.org/wiki/Nonogram>

We maken een spel waarbij de gebruiker nonogrammen moet oplossen in oplopende graad van moeilijkheid.

PROGRAMMAVEREISTEN

Als de gebruiker het spel start moet hij een gebruikersnaam en paswoord ingeven. Er wordt gecontroleerd in een (eenvoudig) geëncrypteerd bestand of deze gebruiker al bestaat of niet. Een nieuwe gebruiker wordt toegevoegd en start met de eenvoudigste opgave. Een gebruiker die al bestond komt automatisch terug terecht op de plek waar hij zich de laatste keer in het spel bevond.

Het spel bevat een aantal nonogrammen (minstens 10, zoek online naar opgaven, je vindt er heel veel) in oplopende graad van moeilijkheid en grootte. We tonen eerst de eenvoudigste opgave. De gebruiker kan nu vakjes inkleuren en vakjes aanduiden die zeker niet ingekleurd mogen worden. Die laatste zijn enkel belangrijk voor de speler zelf als hulp. Wanneer de speler alle juiste vakjes ingekleurd heeft (en geen extra!) wordt dit automatisch door het spel gedetecteerd, krijgt hij felicitaties en wordt de oplossing benoemd (voor opgave hierboven bijvoorbeeld: "Proficiat, een kat!"). Daarna gaat hij naar een volgende opgave (die dan weer een beetje moeilijker is.) De opgaven komen uiteraard uit een bestand en zitten dus niet hard-coded in je applicatie. Deze bestanden zijn ook best (eenvoudig) geëncrypteerd zodat de speler niet te makkelijk te oplossing kan bekijken.

Is de speler het even beu, dan kan hij besluiten om het spel af te breken. Dat kan via een menu-item (afsluiten) of gewoon door de applicatie te stoppen. Sowieso wordt op elk moment

de voortgang van de speler op schijf bijgehouden: in welke level zit hij en wat is zijn huidige inkleuring van het bord.

Het moet ook mogelijk zijn (via een menu-item) een overzicht te krijgen van alle spelers en tot welke level ze al gevorderd zijn, in volgorde (verst gevorderde staat bovenaan, met ook een indicatie van op welke dag die prestatie geleverd werd). Bovendien moet je in het spel ook van speler kunnen wisselen (door uit te loggen en dan terug in te loggen als andere speler).

Ten slotte is het uiteraard mogelijk om de spelregels te bekijken. Deze komen ook uit een bestand.

UITBREIDINGEN

- Voeg een systeem van sterren toe aan het oplossen van een level. Heb je een level opgelost, dan krijg je een aantal sterren (1, 2 of 3). Dit wordt toegekend aan de hand van het aantal keer dat je een foute pixel inkleurde tijdens het oplossen van de puzzel (nooit een fout: 3 sterren, soms 2 sterren, vaak 1 ster).
 - Heb je het sterrensysteem geïmplementeerd, zorg er dan voor dat de speler een level opnieuw kan spelen om toch de 3 sterren te halen. Het aantal sterren wordt meegeteld in het bepalen van de ranking in het overzicht van de spelers.
 - Voeg een timing toe aan de oplossing: je houdt bij hoe lang de speler doet over een bepaalde opgave. Opgelet: dit wil zeggen dat je continu zijn tijd moet bijhouden, want de speler kan op elk moment de applicatie afbreken, en dan moet je zijn tijd tot dan toe opgeslagen hebben! Als de speler nadien zijn spel terug opstart loopt de tijd gewoon verder...
 - Advanced: schrijf een opgave generator, die een (random) opgave genereert. Je kan niet zomaar een random verzameling pixels nemen en daarvan de bijbehorende getallen als opgave nemen. Niets garandeert je dat die opgave een unieke oplossing heeft!
-