

1. INHOUD

Bestanden uitlezen, bestanden schrijven en data toevoegen aan een bestand, csv-bestanden

2. OEFENINGEN

- Demo 1: Notepad
- Demo 2: Read CSV-file
- Demo 3: Write CSV-file
- A: Plot data
- A: Draw map
- A: Map Editor
- E: Draw and update scene
- X: Sudoku from file

2.1 Demo 1: Notepad

Schrijf een programma die toelaat om een bestand uit te lezen en weer te geven in een tekstbox. Een gebruiker kan aanpassingen maken aan de tekst in deze tekstbox en dit dan achteraf terug opslaan in hetzelfde of een ander bestand. Maak hierbij gebruik van de filedialog van C#.

2.2 Demo 2: Read CSV-file

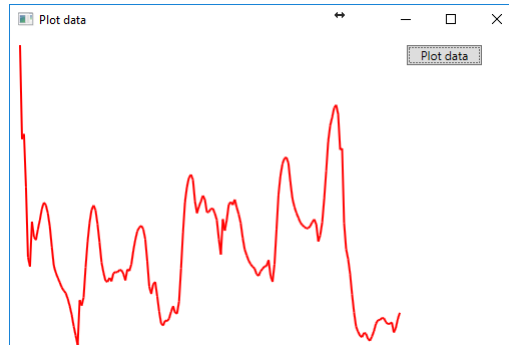
Het CSV-bestandsformaat is een gekend formaat omdat het toelaat om data in een geordende wijze op te slaan. Alle waarden worden hierbij door een gegeven teken van elkaar afgezonderd (typisch de komma). Lees het CSV-bestand in bijlage in en print alle waarden af in een textbox in tabelvorm. Bereken ook het gemiddelde van de waarden.

2.3 Demo 3: Write CSV-file

In deze laatste demo wordt een CSV-bestand aangemaakt a.d.h.v. 2 arrays gevuld met randomwaarden. Genereer de array en sla deze op in het bestand, waarbij de waarden gescheiden worden door een komma.

2.4 A: Plot data

In bijlage kan je het bestand “history_export_2017-12-12T08_25_38.csv” terugvinden. Deze CSV bevat een aantal kolommen met gemeten waarden. Zorg ervoor dat je de temperatuur kan plotten zoals weergegeven in figuur 1. Zorg ervoor dat de kleinste waarde uit de kolom overeenkomt met de onderkant van je canvas en de grootste waarde overeenkomt met de bovenkant van de canvas. Schall ook de waarden volgens de horizontale as uit t.o.v. de canvas. Voer de nodige converties uit om van string (eventueel string replace) naar double over te gaan zodat de waarden geplotted kunnen worden.



Figuur 1: Plotten van de temperatuur in een grafiek.

2.5 A: Draw map

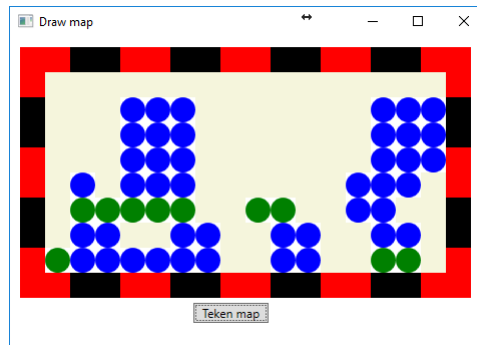
Teken de map vanuit het bestand “map.text” zoals afgebeeld uin figuur 2. Maak hierbij ook een struct aan waarin je volgende elementen opneemt:

- de kolom (int),
- de rij (int),
- het type element (char)

Lees eerst het bestand in en sla alle tekens in een aparte struct op. De positie van elk karakter in het bestand bepaalt ook de kolom- en rij-indices. Sla alle structs op in een lijst en geef daarna deze lijst door aan een tekenfunctie. Het tekenen van de tekens uit het bestand verloopt als volgt:

- een '+' wordt voorgesteld door een rood gevulde vierkant,
- een '-' wordt voorgesteld door een zwart gevulde vierkant,
- een '0' wordt voorgesteld door een beig gevulde vierkant,
- een '1' wordt voorgesteld door een blauw gevulde schijf,
- een '2' wordt voorgesteld door een groen gevulde schijf.

Alle figuren zijn hierbij 25 bij 25 pixels groot.



Figuur 2: Plotten van de temperatuur in een grafiek.

2.6 A: Map Editor

Herneem de voorgaande opgave, maar zorg er nu voor dat je een willekeurig element van de map kan aanpassen volgens 1 van de 5 mogelijk opties. Gebruik hiervoor een combobox om zowel het element en het type te selecteren.

2.7 E: Draw and update scene

Analyseer de CSV “figures.data”. Zorg ervoor dat je de figuren kan tekenen zoals aangegeven in de CSV. Het formaat van de CSV is als volgt:

- Kolom 1: type figuur (rechthoek, cirkel of lijn),
- Kolom 2: positie x1,
- Kolom 3: positie y1,
- Kolom 4: positie x2,
- Kolom 5: positie y2,
- Kolom 6: kleur in grijswaarde, tussen 0 en 255.

Neem heerbij een canvas van 400 bij 400 pixels om alle figuren te kunnen tekenen.

Updaten

Kan je deze oplossing aanpassen zodat je de figuren kan aanpassen, figuren kan toevoegen en figuren kan verwijderen?

2.8 X: Sudoku from file

In dit programma wordt een sudoku-solver geprogrammeerd. De gebruiker kan een sudoku vanuit een bestand importeren en de oplossing van de sudoku (indien bestaande) terug exporteren naar een oplossingenbestand (kies CSV-formaat). Indien de oplossing bestaat wordt de oplossing in hetzelfde formaat als de input opgeslagen. Indien niet wordt in het resultaat-bestand gemeld dat de sudoku niet opgelost kon worden. De keuze van de bestanden wordt aan de gebruiker overgelaten via filedialogs.

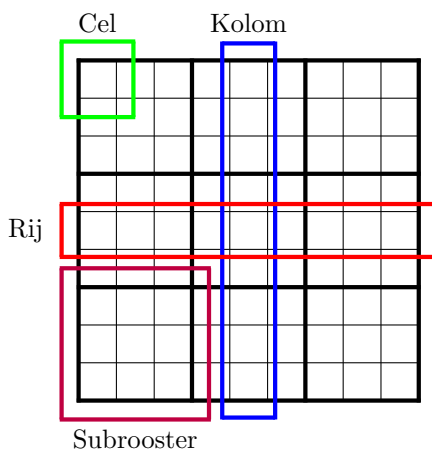
Werkwijze sudoku

De klassieke Sudoku bestaat uit een rooster (Figuur 3) van 9 bij 9 cellen (81 elementen), waarbij volgende spelregels gehanteerd worden:

- in elke rij is elk getal uniek
- in elke kolom is elk getal uniek,
- elk subrooster van 3 bij 3 cellen bevat 9 unieke getallen.

De werkwijze van backtracking wordt hieronder beschreven.

- Ga naar de eerste lege cel in de Sudoku.
- Probeer de laagste waarde, hier 1, in te vullen. Ga naar de volgende lege cel indien aan alle spelregels voldaan zijn. Indien niet, incrementeer de celwaarde en controleer opnieuw. Blijf dit herhalen totdat een mogelijke kandidaat gevonden is.
- Indien bij een verdere cel de maximale waarde overschreden wordt (hier 9), moet het algoritme terugkeren naar de vorige ingevulde cellen. Die cellen worden opgehoogd en het algoritme rekt opnieuw verder. Indien er geen oplossingen mogelijk zijn, zal het algoritme uit zichzelf opnieuw naar de oorsprong begeven, waar het stopt met een foutmelding.



Figuur 3: Schematische voorstelling van een 9 bij 9 sudoku.

Programmeer een functie die toelaat om dergelijke puzzels op te lossen. Je mag hierbij sudokus uit een krant nemen en hardcoded programmeren in het programma.