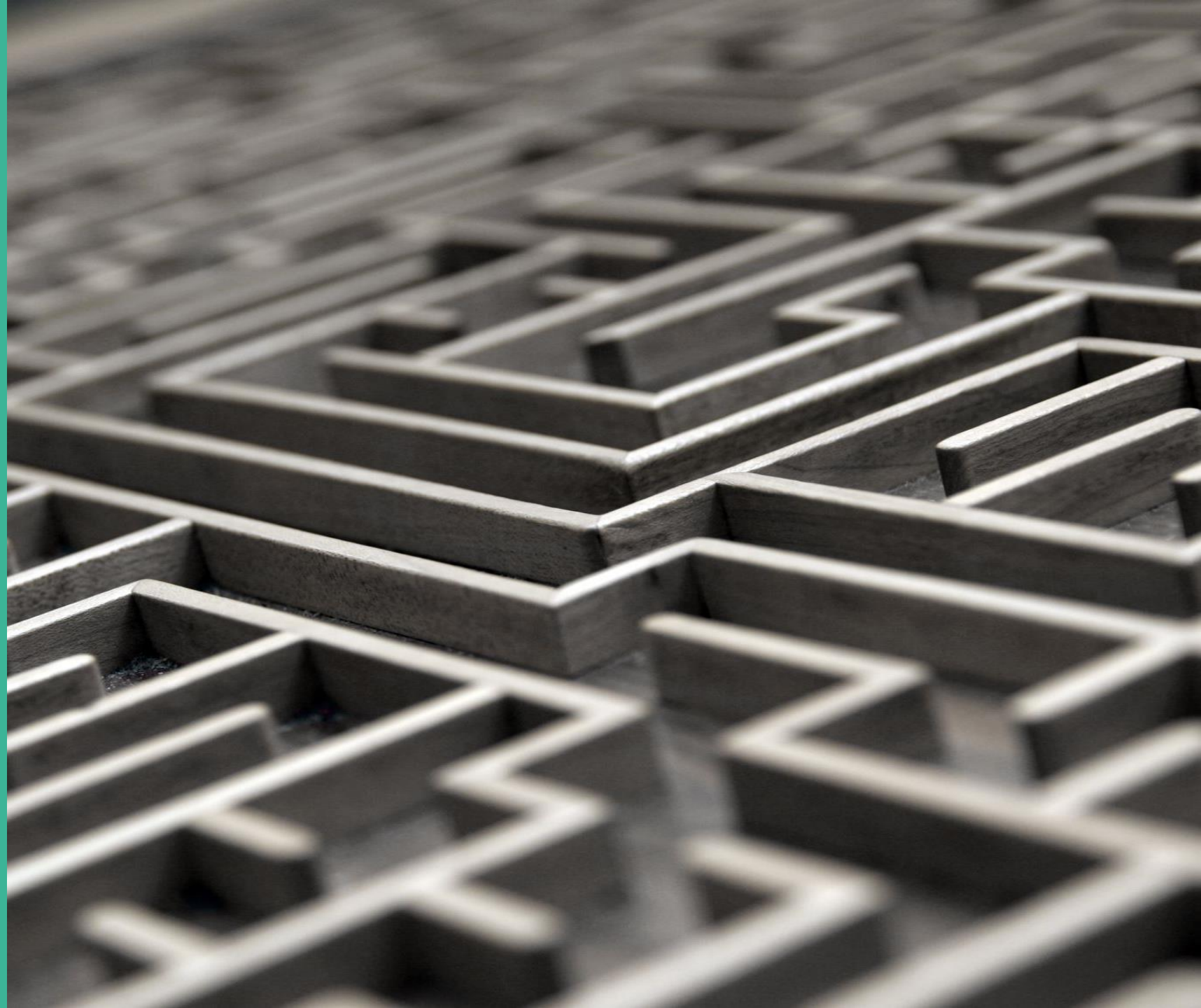
Opdracht 1: Maze generation

Applied Programming

Tim Vermeulen

26/9/2023

Maze generation



Odisee
DE CO-HOGESCHOOL

1. Theorie

Doolhof

- Verschillende vormen mogelijk
- Begin en einde (of meerdere)
- Eén of meerdere wegen tussen begin en einde
- Posities discreet of continue
- Muren met vaste dikte of variabel
  - Gebieden die onbereikbaar zijn
  - Grotere stukken muur die een groter gebied in nemen
- Een ‘perfect’ doolhof
  - Tussen 2 punten exact 1 pad
  - Elk punt bereikbaar

Maze generation

Algoritmen gebruikt om een oplosbaar doolhof op te stellen

Muren slopen

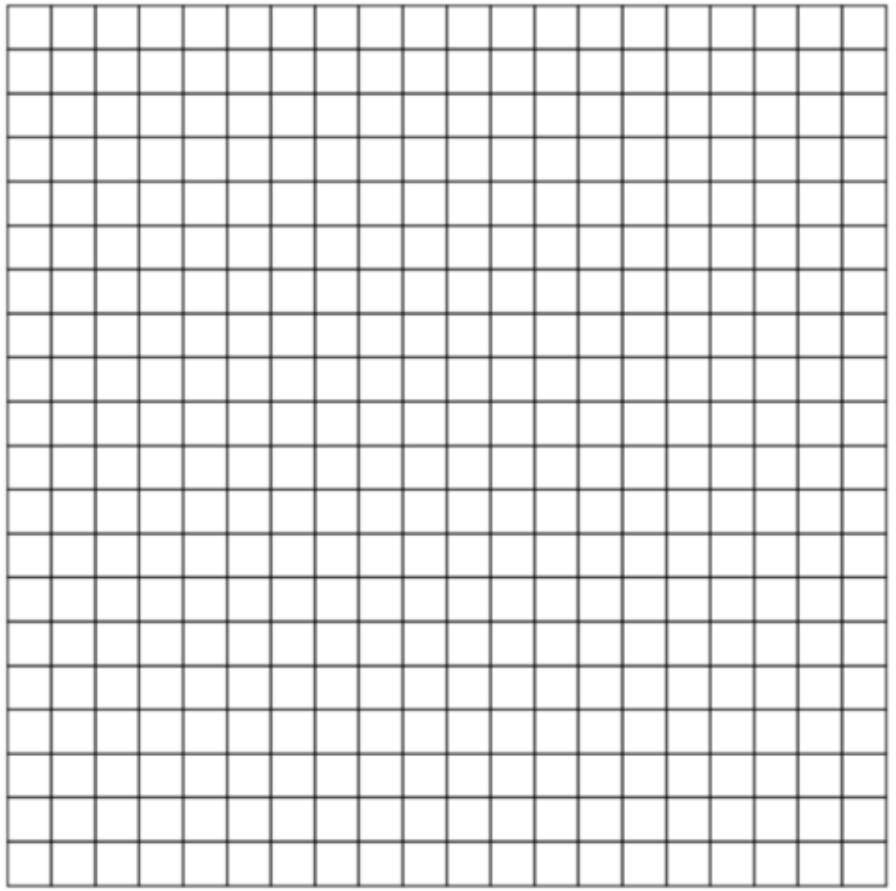
- Vertrekken van een wereld vol muren
- Sloop muren
  - Willekeurig, maar met een systeem
  - Tot elke plaats bereikbaar wordt
- Algoritme levert pad op

Muren toevoegen

- Vertrekken van een lege wereld met enkel zijkanten
- Voeg muren toe
  - Willekeurig, maar met een systeem
  - horizontale en verticale muren toevoegen
  - Stoppen na een bepaald aantal iteraties
- Nadien pad bepalen

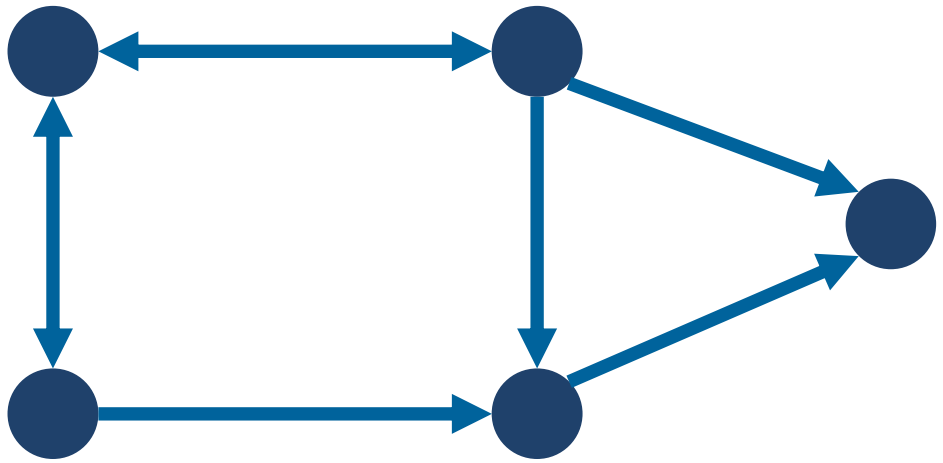
Muren slopen

- Een matrix van cellen
- Elke cel heeft 4 muren
- Elke cel is verbonden met buren
- Deze verbindingen kunnen voorgesteld worden als een *graph*
- Verschillende algoritmen doorlopen deze *graph* en zetten deze om naar een *spanning tree*



Graph

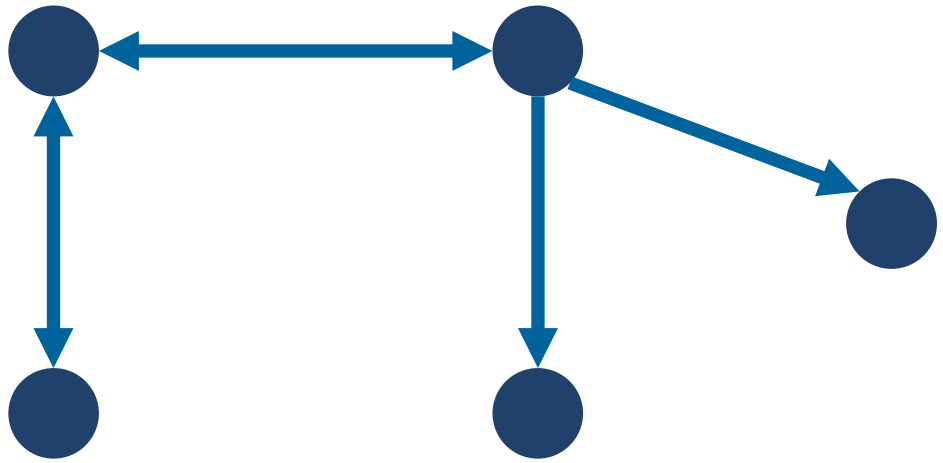
- Datastructuur die meerdere punten/knopen met elkaar verbindt
  - Knopen = vertices
  - Verbindingen = *edges*
  - Kan richting hebben
    - Directioneel / niet directioneel
    - In de figuur rechts, beiden
    - Bij doolhof niet directioneel



- Bij een doolhof
  - Knopen zijn de cellen
  - Cellen zijn verbonden als er geen muur tussen staat

Spanning tree van een *graph*

- Bevat alle knopen van de *graph*
- Bevat enkele (of eventueel alle) verbindingen van de *graph*
- Alle punten verbonden
- Minimum spanning tree
  - Alle punten maar op 1 manier bereikbaar



Doolhof opstellen met spanning tree

- De graph om van te starten bevat alle cellen
  - Alle rechtstreekse buren zijn verbonden
- Random minimum spanning tree bepalen van deze graph
  - Verschillende algoritmes beschikbaar
- Rechtstreekse buren niet verbonden in de tree
  - Muur tussen plaatsen

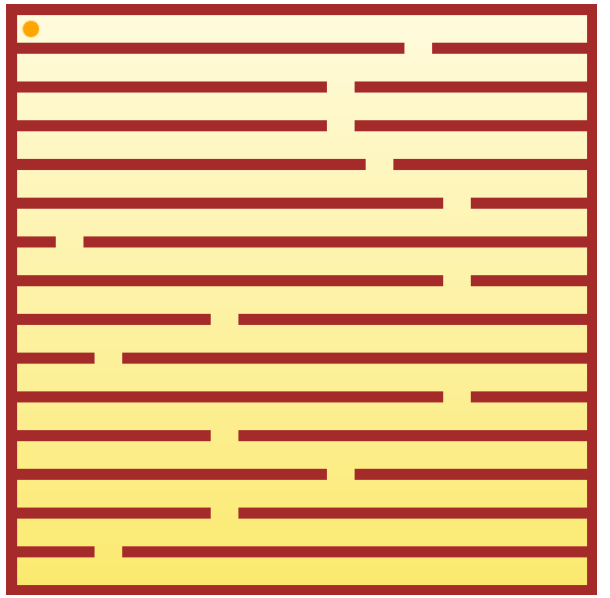
- Resultaat: een ‘perfect’ doolhof

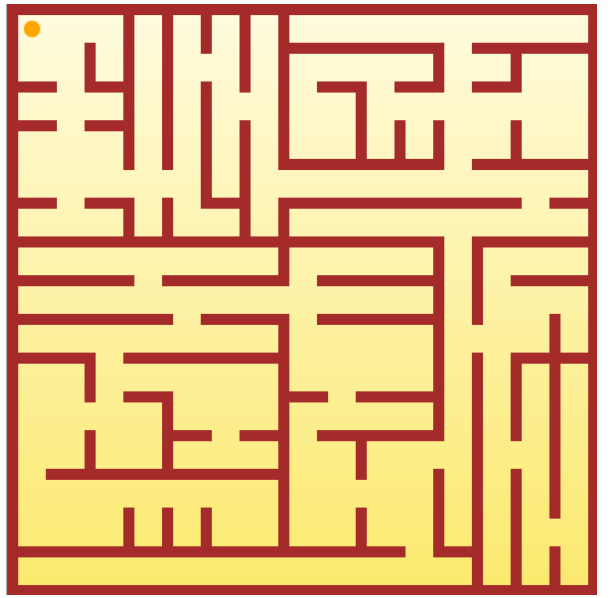
Voorbeeld algoritme: Depth first search

- Begin bij een cel naar keuze
- Herhaal volgende stappen:
  - Duid de huidige cel aan als bezocht
  - Kies een willekeurige buur van de huidige cel die nog niet bezocht is
  - Verwijder de muur tussen deze cel en de buur
  - Neem de gekozen buur als nieuwe ‘huidige cel’
- Als een cel geen onbezochte buren heeft, ga dan terug (back-track) op je pad van bezochte cellen, tot je een cel tegen komt met onbezochte buren. Dit wordt dan je nieuwe ‘huidige cel’
  - Je zal dus een stack moeten bijhouden van je laatst bezochte cellen
- Als je terug bij je start cel komt, heb je alle cellen bezocht

Muren toevoegen

- Vertrekken van een lege wereld met enkel zijkanten
- Muren toevoegen
  - Kan met een vast grid of vaste afstand tussen muren
  - Kan met nog meer willekeurige patronen
- Er zijn eenvoudige algoritmes zelf te verzinnen
  - Enkel horizontale en/of verticale muren,
    met op willekeurige plaatsen een doorgang
    - Steeds een oplossing gegarandeerd





2. Opdracht

Opdracht

- Stel een model op voor je doolhof
- Voorzie 3 manieren om een doolhof te genereren
- Toon het doolhof in een 2D WPF applicatie
- Experimenteer en documenteer

Model

- Stel een model op om een doolhof bij te houden
  - 2D coördinaten
  - Onafhankelijk van grafische frameworks (GDI, WPF, …)
  - Onafhankelijk van je algoritmen om een doolhof te genereren
  - Een doolhof met
    - muren met een bepaalde dikte
    - een bal

Implementeer 3 manieren om een doolhof te genereren

1

Statisch

Een opsomming van aan te maken muren in code of uit een bestand

2

Muren toevoegen

Vertrek van een lege wereld en gebruik een eenvoudig algoritme om muren toe te voegen

3

Muren slopen

Gebruik een algoritme zoals gezien in de slides

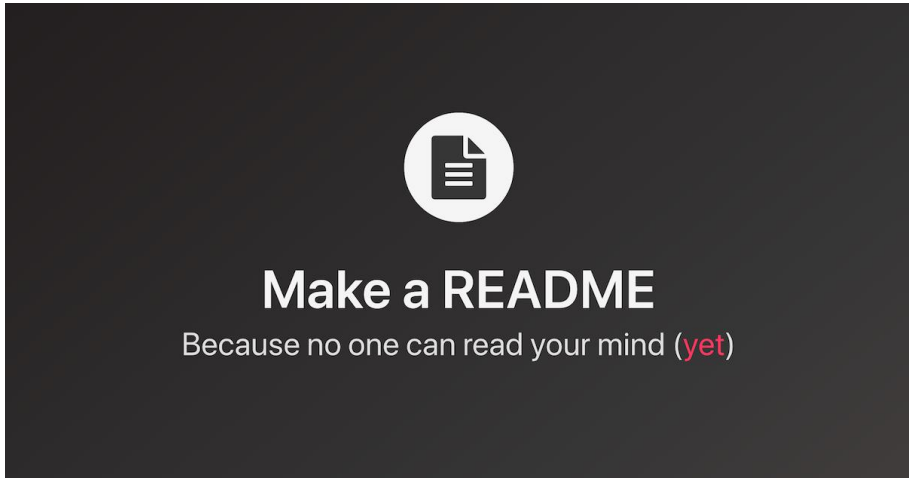
Structuur

- Zorg voor een factory model
  - Elke manier om een doolhof te genereren zit in een factory
  - Nieuwe factories toevoegen gaat eenvoudig
- Gebruik een interface en overerving
  - Factories aanspreken gaat op dezelfde manier
  - Alle factories genereren een doolhof met hetzelfde datamodel

2D weergave in WPF applicatie

- Maak een 2D weergave in een WPF applicatie
  - Zorg dat je het doolhof grafisch kan weergeven/tekenen
    - Muren + bal
  - Bij de start van de applicatie wordt een eerste doolhof getoond
  - Er kan in de gui de keuze gemaakt worden uit één van de 3 manieren om een doolhof te genereren
  - Met een knop kan telkens een nieuw doolhof gegenereerd worden
    - De applicatie moet niet herstart worden

Documentatie



- readme.md
  - 1 bestand/verslag voor het hele vak, hoofdstuk per opgave
  - Beschrijving structuur projecten
  - Beschrijving van model
  - Beschrijving van algoritmen
    - Algemene theoretische werking
    - Technische keuzes en oplossingen
  - Bespreking van werking grafische applicatie + screenshot
  - Algemene bedenkingen / reflectie