Intelligente Pfadsuche

Vergleichende Simulation von Suchverfahren in generierten Maze-Umgebungen

Ole Matzky, 21.07.2025

Angewandte Modellierung und Systemsimulation

Gliederung

01 Einleitung

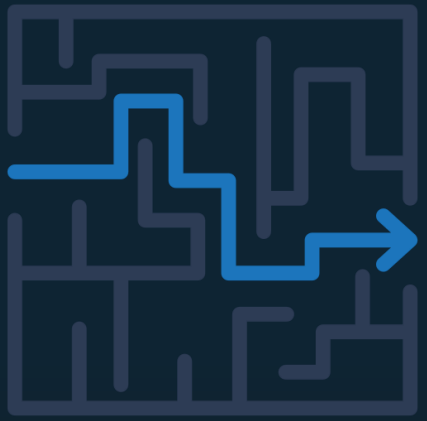
02 Maze Generiung

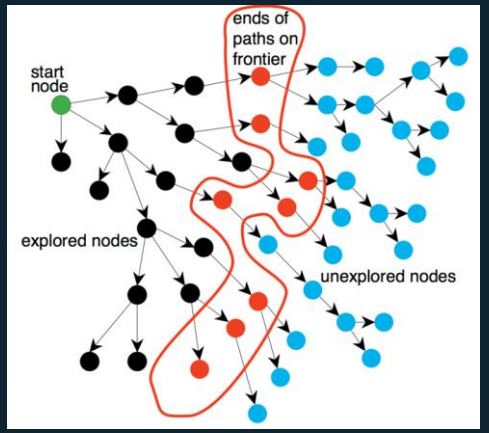
03 GUI

04 Algorithmen

01 Einleitung

Einleitung




ends of
paths on
frontier
start
node
explored nodes
unexplored nodes


PY

02

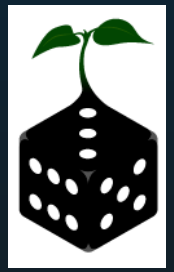
Maze Generierung

Maze Generierung

- NumPy NDArray

- Randomized DFS

- Random Seed

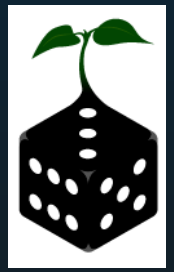


Maze Generierung

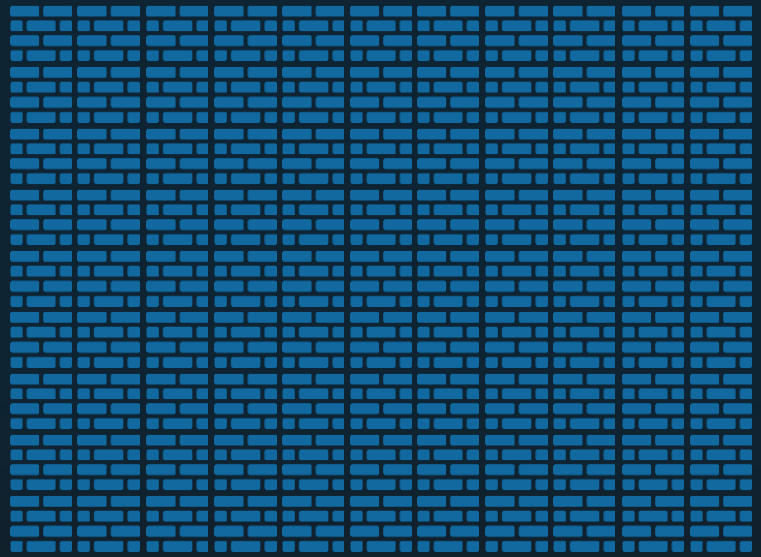
- NumPy NDArray

- Randomized DFS

- Random Seed

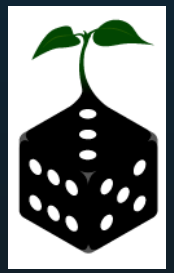


9x11

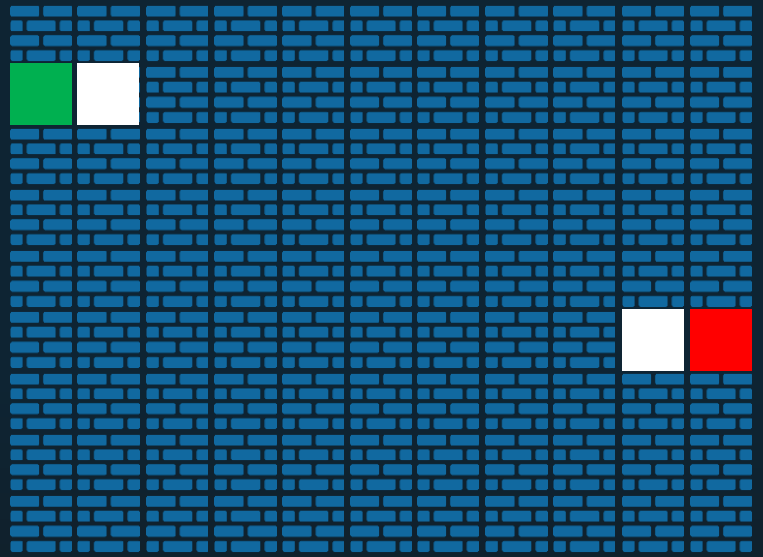


Maze Generierung

- NumPy NDArray
- Randomized DFS
- Random Seed



9x11

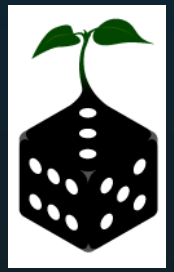


Maze Generierung

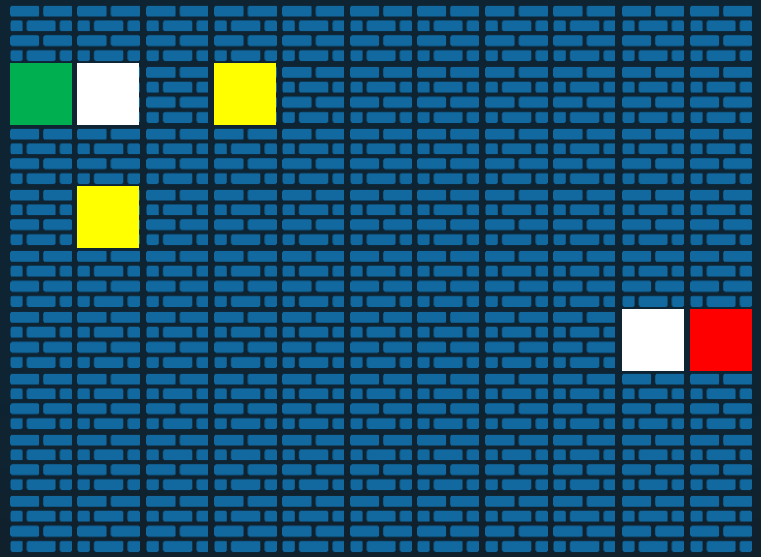
- NumPy NDArray

- Randomized DFS

- Random Seed



9x11

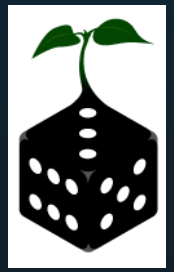


Maze Generierung

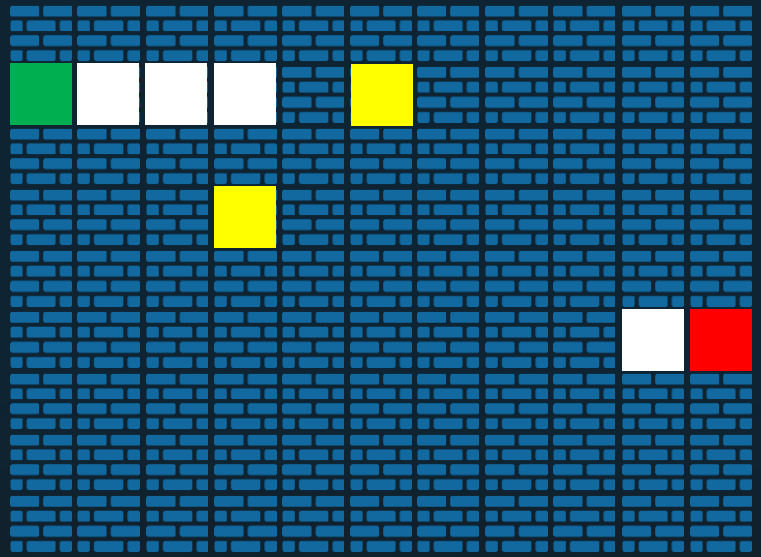
- NumPy NDArray

- Randomized DFS

- Random Seed



9x11

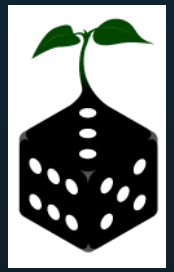


Maze Generierung

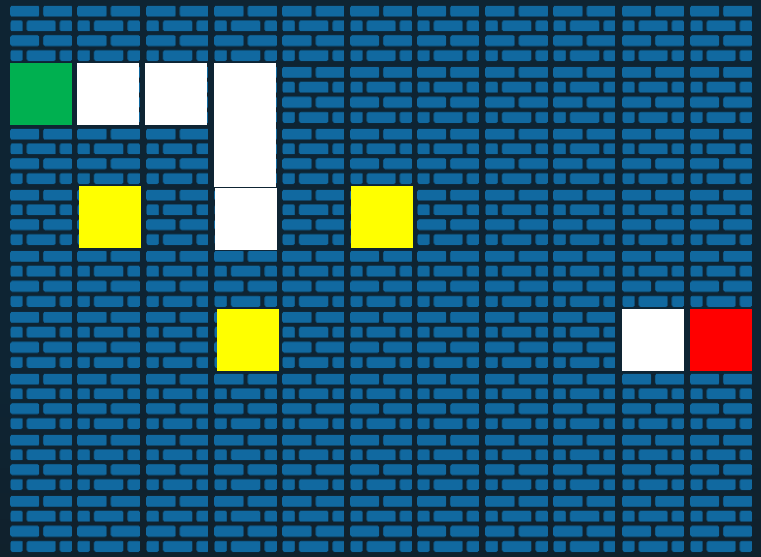
- NumPy NDArray

- Randomized DFS

- Random Seed

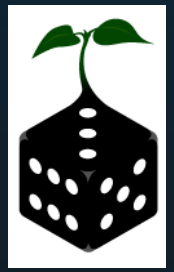


9x11

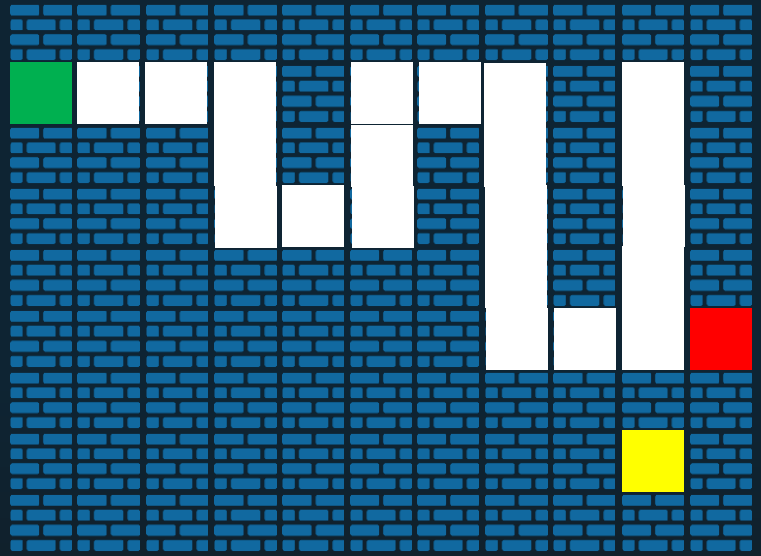


Maze Generierung

- NumPy NDArray
- Randomized DFS
- Random Seed



9x11



03

GUI

GUI

Tkinter

Intelligent Pathfinding: Comparative Simulation of Search Strategies in Generated Maze Environments

Maze Parameters

Width
70

Height
50

Extra openings (%)
7

Animate Generation

Random Seed

Seed: 42

Algorithms

A*

Dijkstra

Greedy BFS

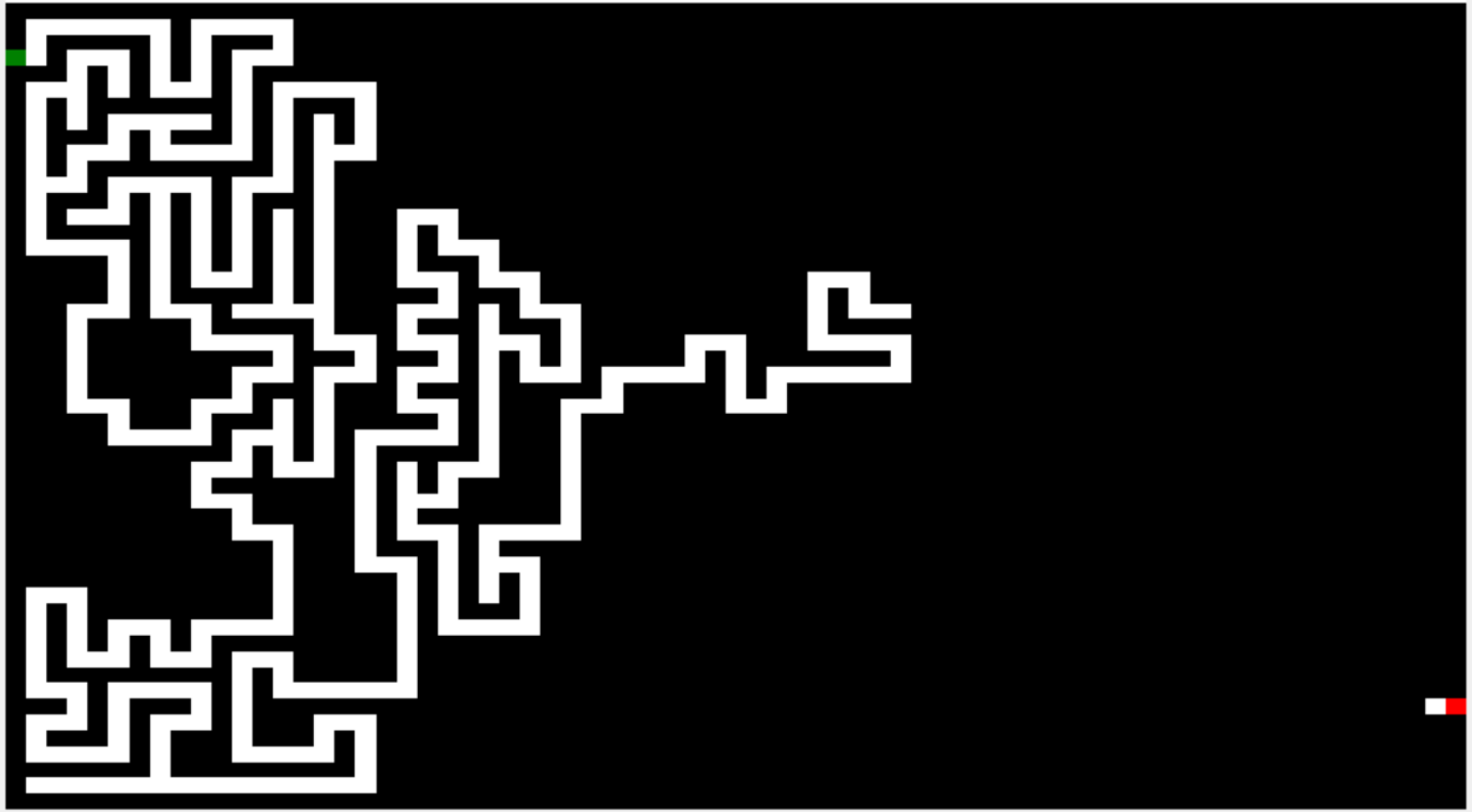
Controls

Delay (ms)
50

Start Animation

Save Animation (MP4)

Maze Generation



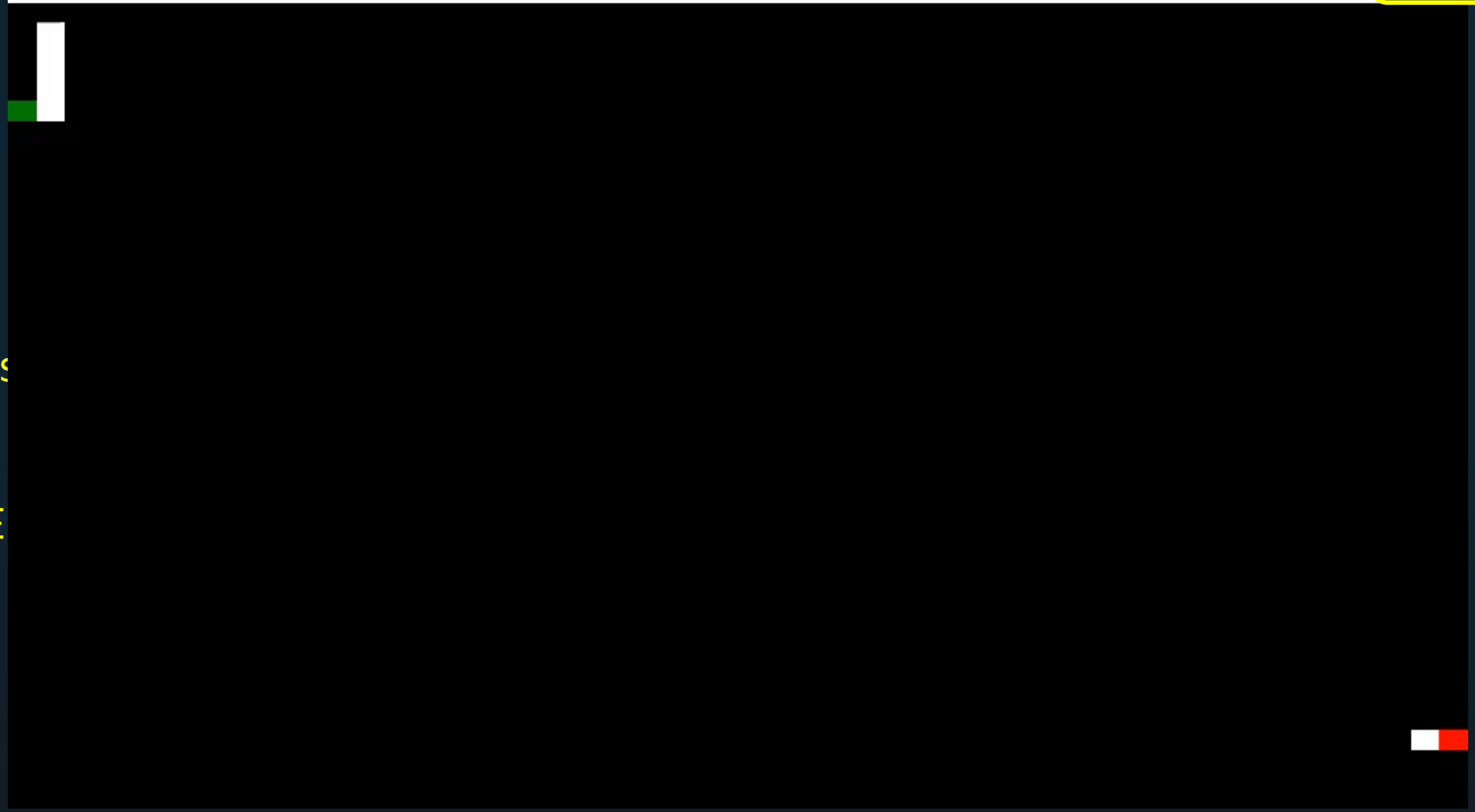
04

Algorithmen

Algorithmen

	A*	Dijkstra	Greedy BFS
Kostenfunktion	f(n)=g(n)+h(n)	f(n)=g(n)	f(n)=h(n)
Eigenschaft	• optimal • mittel schnell	• optimal • langsam	• nicht optimal manchmal • sehr schnell

Maze Generation



4x

THANKS

- Search Algorithm Icon
- Numpy Icon
- Mein Githup-Repo
- Matplotlib Icon
- Random Seed Icon

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik** and illustrations by **Stories**