

Python-Projekt: Sudoku Solver

Leander Teichmann, Florian Herrmann

8. März 2021

Python-
Projekt:
Sudoku Solver

Leander
Teichmann,
Florian
Herrmann

Was ist ein
Sudoku?

Projektaufbau

Was ist
Rekursion?

Lösungsstrate-
gie

GUI

Fazit

1 Was ist ein Sudoku?

2 Projektaufbau

3 Was ist Rekursion?

4 Lösungsstrategie

5 GUI

6 Fazit

- Das Spielfeld besteht aus $n \cdot n$ Feldern
- In der Regel: $n = 9$
- Ziel: Alle leeren Felder mit Zahlen füllen, sodass die Zahlen von 1 - 9 jeweils nur einmal vorkommen
- in jeder Spalte
- in jeder Reihe
- in jedem der neun kleineren Quadrate

	6				5		1	
	5			3	1		9	
3			9					4
								6
2	3	9				8	4	1
6								
8					3			2
	7		4	6			8	
	9		1				7	

Python-
Projekt:
Sudoku Solver

Leander
Teichmann,
Florian
Herrmann

Was ist ein
Sudoku?

Projektaufbau

Was ist
Rekursion?

Lösungsstrate-
gie

GUI

Fazit

Ziele:

Muss:

Sudoku-Solver

Soll:

Dokumentation

Variable Sudoku-Gittergröße

Kann:

GUI

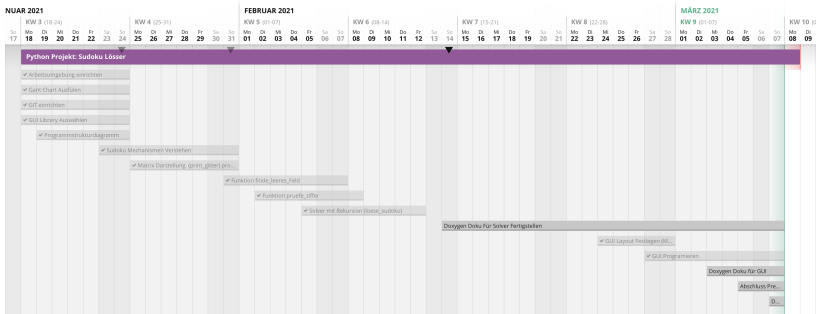
Versionsverwaltung mit git

Organisation mit Agantty (Ganttchart)

Python- Projekt: Sudoku Solver

Leander
Teichmann,
Florian
Herrmann

Ganttchart:



Was ist ein
Sudoku?

Projektaufbau

Was ist
Rekursion?

L sungsstrate-
gie

GUI

Fazit

Rekursion ist ein Programmierkonzept, bei der eine Funktion nur einen kleinen Teil des Problems löst und damit ein Problem ein bisschen verkleinert und sich dann selbst aufruft, um den Rest des Problems zu lösen.

Das wird so lange fortgesetzt, bis das Problem gelöst ist.

```
def fibo(n):  
    if n <= 1:  
        return n  
    else:  
        return (fibo(n - 1) + fibo(n - 2))  
  
nterms = 10  
print("Fibonacci sequence:")  
for i in range(nterms):  
    print(fibo(i))
```

Fibonacci sequence:

0
1
1
2
3
5
8
13

Python-
Projekt:
Sudoku Solver

Leander
Teichmann,
Florian
Herrmann

Was ist ein
Sudoku?

Projektaufbau

Was ist
Rekursion?

Lösungsstrate-
gie

GUI

Fazit

- Backtracking (deutsch: Rücksetzverfahren) bezeichnet eine Problemlösungsmethode innerhalb der Algorithmik.
- Trial-and-Error-Prinzip

Backtracking beim Sudokulösen:

1. Suchen eines leeres Feldes
2. Versuchen, die Ziffern 1 - 9 an dieser Stelle zu platzieren
3. Prüfen anhand des aktuellen Gitters, ob diese Ziffer an der aktuellen Stelle gültig ist
 - a. Wenn die Ziffer gültig ist: Versuchen das Gitter rekursiv mit den Schritten 1 - 3 zu füllen.
 - b. Wenn sie nicht gültig ist: Setzen des gerade gefüllten Feld auf 0 und zurückgehen zum vorherigen Schritt.

Ist das Gitter voll, wurde eine Lösung gefunden.

Python-
Projekt:
Sudoku Solver

Leander
Teichmann,
Florian
Herrmann

Was ist ein
Sudoku?

Projektaufbau

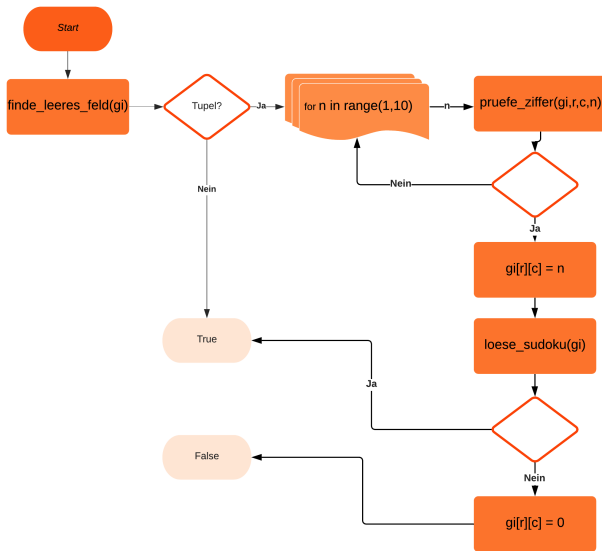
Was ist
Rekursion?

Lösungsstrate-
gie

GUI

Fazit

Flowchart des Lösungsalgorithmuses:



GUI mit Pygame

Spielfeld besteht aus verschiedenen Instanzen von Klassen:
Felder

- X Position
- Y Position
- Inhalt des Feldes

Button

- X Position
- Y Position
- Breite
- Höhe
- Beschriftung

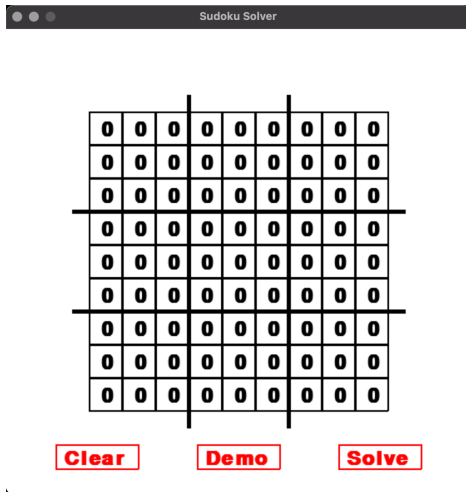
Funktionen von Felder

- Init
- Draw
- Select

Funktionen von Button

- Init
- Draw
- Select

Fertiges GUI:



Python-
Projekt:
Sudoku Solver

Leander
Teichmann,
Florian
Herrmann

Was ist ein
Sudoku?

Projektaufbau

Was ist
Rekursion?

Lösungsstrate-
gie

GUI

Fazit

Muss:

Sudoku-Solver \Rightarrow Mittels Backtracking

Soll: Dokumentation \Rightarrow Docstrings

Variable Sudoku-Gittergröße \Rightarrow In der Library implementiert
Ab 16x16 großer Rechenaufwand. In der Komplexität reduzierte
16x16-Gitter

können in kurzer Zeit gelöst werden.

Kann:

GUI \Rightarrow Mittels Pygame

Learnings:

Konzepte Rekursion und Backtracking

Pygame

Verbesserung von Projektorganisationsstrukturen

Kennenlernen von git

Code und Präsentation auf <https://github.com/flowlow969/TheSuperFancySudokuSolver>.