BOSCH
Technik fürs Leben

DHBW
Duale Hochschule
Baden-Württemberg

Sudoku Helper

Studienarbeit

Bachelor of Science

des Studiengangs Informatik

an der Dualen Hochschule Baden-Württemberg Stuttgart

von

Ruben Hartenstein, Annika Harter

10.06.2022

Bearbeitungszeitraum	22.10.2022 - 10.06.2022
Matrikelnummer, Kurs	2746235, 4810277, TINF19ITA
Ausbildungsfirma	Robert Bosch GmbH, Stuttgart
Betreuer	Sebastian Trost
Gutachter	Sebastian Trost

Selbstständigkeitserklärung

Ich versichere hiermit, dass ich meine Studienarbeit mit dem Thema: *Sudoku Helper* selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Stuttgart, 10.06.2022

Ruben Hartenstein, Annika Harter

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

Abstract

TODO: deutscher Abstract....

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

Inhaltsverzeichnis

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

Abkürzungsverzeichnis

BRC	Block-Reihe-Check
BSD	Berkley Software Distribution
CE	Clear everything
CSS	Cascading Style Sheets
HTML	Hypertext Markup Language
HTTP	Hypertext Transport Protocol
HTTPS	Hypertext Transport Protocol Secure
MVC	Model View Controller
RBC	Reihe-Block-Check
UI	User Interface
UX	User Experience

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

Abbildungsverzeichnis

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

Tabellenverzeichnis

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

Formelverzeichnis

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

Listings

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

1. Einleitung

In vielen Zeitungen ist eine Vielzahl von Logikrätseln gedruckt. Darunter eines der häufigsten Logikrätsel: das Sudoku. In der üblichen Version ist es das Ziel, ein 9×9-Gitter mit den Ziffern 1 bis 9 zu füllen. Dabei kann das Gitter in drei unterschiedliche Einheiten aufgeteilt werden. Diese Einheiten sind die Spalten, Zeilen und Blöcke des Gitters. Ein Block ist ein 3×3-Unterquadrat des Gitters. In jeder Einheit darf jede Ziffer genau einmal vorkommen. Ausgangspunkt ist ein Gitter, in dem bereits mehrere Ziffern vorgegeben sind, damit das Sudoku eindeutig lösbar ist. Sudokus gibt es in unterschiedlichen Schwierigkeitsgraden.

In der Studienarbeit soll ein **Sudoku Helper** entwickelt werden. Ziel ist es dem Nutzer verschiedene Strategien unterschiedlicher Schwierigkeitsstufen anhand eines realen Beispiels zu erklären und so den Lösungsweg für schwere Sudokus zu erläutern. [Zei12] [Sud22]

Im Verlaufe der Arbeit wird zunächst die Motivation und Aufgabenstellung der Studienarbeit beschrieben.

Danach werdend die Rahmenbedingungen, wie beispielsweise die Wahl der Programmiersprache, des Frameworks und des Webservers erläutert. Im Zug dieser wissenschaftlichen Vertiefung wird auch genauer hinter die Mathematik die hinter Sudokus steht, geschaut. Dabei werden verschiedene Themen wie die Komplexität eines Sudokus oder die eindeutige Lösbarkeit diskutiert und erläutert.

Der nächste größere Teil der Studienarbeit ist die Implementierung. Hierbei werden Designentscheidungen für Front- und Backend erläutert, sowie deren technische Umsetzung. Im Zug der Implementierung werden auch die Testing- und Validierungsstrategien beschrieben.

Am Schluss der Arbeit steht ein Fazit mit den Ergebnissen und Erfahrungen, sowie ein Ausblick wie das Projekt noch verbessert und erweitert werden könnte.

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

1.1. Motivation

Sudokus gibt es mittlerweile in vielen verschiedenen Varianten, mit modifizierten Regeln. In dieser Arbeit und für den **Sudoku Helper** wird jedoch nur das klassische Sudoku, wie es in der Einleitung beschrieben wurde, betrachtet. Es gibt verschiedene Schwierigkeitsstufen von Sudokus. Meistens werden Sudokus mit vielen vorgegebenen Zahlen als einfach eingestuft und Sudokus mit wenigen vorgegebenen Zahlen als schwierig. Ein weiterer Punkt, der Einfluss auf die Schwierigkeitsstufe eines Sudokus hat, ist die Anordnung der vorgegebenen Ziffern.

Wer schon einmal versucht hat ein schweres Sudoku zu lösen, ist vielleicht auch bereits an seine Grenzen gekommen. Gerade bei schwierige Sudokus mit wenigen vorgebenden Zahlen kann es schwierig sein selbst eine passende Lösungsstrategie für das Sudoku zu finden. Oft sind anwendbare Lösungsstrategien schwierig zu erkennen und es lassen sich damit meist nur Kandidaten eliminieren. Einfachen Sudokus lassen sich oft durch simple logische Schlussfolgerungen lösen und Lösungsstrategien werden meist unbewusst und intuitiv eingesetzt. [Mar]

In dieser Studienarbeit soll ein **Sudoku Helper** entwickelt werden, der Lösungsstrategien programmatisch erkennt und dem Benutzer bei dem Finden eines Lösungsschrittes unterstützt. Diese Unterstützung erfolgt dabei in mehreren Stufen, von der Visualisierung, über die Erklärung bis zur automatischen Umsetzung der Strategie.

1.2. Forschungsstand

Im Internet findet man immer wieder Websites mit dem Angebot Sudokus zu lösen. Hierbei können die angegebenen Zahlen eingegeben werden und das Sudoku wird korrekt vervollständigt, falls es eine eindeutige Lösung gibt. Dabei bekommt der Nutzer nur das gelöste Sudoku ohne Informationen darüber, wie man ein Sudoku als Mensch ohne die Rechenleistung eines Computers logisch lösen kann. Viele Websites erklären bereits, welche Lösungsstrategien existieren, wie sie funktionieren und wie sie sich anwenden lassen, nutzen dabei aber meist konkrete Beispiele.

Es existieren aber auch schon ähnliche Varianten zu dem in der Studienarbeit umgesetzten **Sudoku Helper**. In der Studienarbeit sollen ein Schrittweise Hilfestellung stattfinden, die in den meisten Fällen nicht umgesetzt wurde.

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

1.3. Forschungsfrage

Für diese Studienarbeit ist die Zielsetzung, ein **Sudoku Helper** zu entwickeln. Ziel ist es nicht ein Sudoku automatisch zu lösen, sondern dem Benutzer Schritt für Schritt den Weg zur Lösung aufzuzeigen. Auf dieser Basis wird zunächst geprüft, ob das Sudoku eindeutig lösbar ist. Ist dies der Fall, muss der Nutzer die Möglichkeit haben sich Anweisungen geben zu lassen, wie man weiter vorgehen könnte und welche Lösungsstrategie der Benutzer anwenden kann, damit man beim lösen des Sudokus fortschreiten kann. Auf Basis der bereits eingetragenen Zahlen und den übrig gebliebenen Kandidaten, soll geprüft werden, welche Strategie momentan anwendbar ist. Dem Benutzer soll daraufhin die anwendbare Strategie genannt und anhand des konkreten Sudokus erklärt werden, damit der Benutzer das Sudoku Lösen kann. Diese Anweisungen, im folgenden auch Hilfestellungen genannt, sind in vier Varianten unterteilt.

Für die erste Hilfestellung soll das Programm dem Benutzer die anzuwendende Strategie nennen und den Bereich hervorheben, auf dem die Strategie angewendet werden soll. Wenn diese Hilfestellung nicht ausreichend ist, dann markiert das Programm die konkreten Zellen und Kandidaten, die von der Strategie betroffen sind. Mit der dritten Hilfestellung soll eine Erläuterung gezeigt werden, warum bestimmte Kandidaten laut Strategie gestrichen werden können. Mit der letzten Hilfestellungen löscht das Programm den betroffenen oder mehrere betroffene Kandidaten. Für Ausnahmen, in denen eine Strategie eine Zahl hervorhebt, die eingetragen werden kann, erfolgt das Eintragen der Zahl anstatt das Löschen von Kandidaten.

Um ein eindeutig lösbares Sudoku vervollständigen zu können, soll der **Sudoku Helper** etwa 25 Lösungsstrategien beherrschen.

Das Programm soll auf einem Ubuntu Server mit Apache und Django laufen. Des Weiteren gibt es zusätzlich die Anforderung, dass das Web-Frontend responsive sein soll und immer eine benutzeroptimierte Darstellungen liefert.

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

Teil I.

Wissenschaftliche Vertiefung

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

2. Rahmenbedingungen

Für das Projekt werden verschiedene Rahmenbedingungen festgelegt. Dazu gehört die Programmiersprache, indem der **Sudoku Solver** programmiert werden soll, ein Webserver auf dem das Programm später laufen wird und ein Framework, dass als Gerüst für das Programm auf dem Webserver fungiert. Wichtig ist, dass das Framework auf die Programmiersprache angepasst ist, um Probleme zu vermeiden. Im Zuge der Recherchen wurden verschiedenen Webserver in Betracht gezogen und besonders die Webserver Apache HTTP Server und Nginx Server aufgrund ihrer Beliebtheit miteinander verglichen. Auch bei den Frameworks gab es mehrere Optionen, von denen zwei insbesondere in Betracht gezogen und evaluiert wurden. Im Folgenden werden die Pros und Kontras der genannten Server und Frameworks aufgezählt und die schlussendliche Entscheidung begründet.

2.1. Programmiersprache

In den Folgenden Abschnitten werden die genutzten Programmiersprachen vorgestellt und die Entscheidungen für diese erläutert. Neben Python für das Backend, wird für das Frontend JavaScript mit der Erweiterung von jQuery verwendet.

2.1.1. Python

Für die Implementierung der Strategien und generell im Backend wurde sich für die Programmiersprache Python entschieden. Python ist eine höhere, universelle Sprache und ist als eine sehr vielseitige Programmiersprache anerkannt. Mit Python kann ein objektorientierter, aspektorientierter oder funktionaler Programmierstil umgesetzt werden und die Typisierung funktioniert dynamisch. Des Weiteren wird Python oft bei der Datenanalyse und bei der Verarbeitung großer Datenmenge verwendet. Zusätzlich spielt die Laufzeit zunächst nur eine untergeordnete Rolle und ist nicht sicherheitsrelevant, wodurch bei der Nutzung von Python keine größeren Probleme entstehen sollten. Ein weiterer Vorteil von Python ist die große Standardbibliothek, die falls nötig durch Module erweitert

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

werden kann. Noch ein Vorteil ist die hohe Kompatibilität mit mehreren Frameworks. Im Laufe des Kapitels werden noch zwei Frameworks vorgestellt, die für das Projekt in Betracht gezogen worden.

Für das Backend ist eine objektorientierte Umsetzung vorgesehen, die mit Python umgesetzt werden kann. Zudem ist Python einfach zu lesen und klar strukturiert. Auch für die Umsetzung der Lösungsstrategien bietet sich Python durch Module für mathematische Berechnungen und Datenwissenschaft an.

In diesem Projekt wird mit der Version 3.9 von Python gearbeitet. [Ern20] [Sar20, S. 50 ff.] [Sta]

2.1.2. JavaScript

Im Frontend wurde sich für JavaScript mit jQuery entschieden. JavaScript ist eine der meist verwendeten Programmiersprachen der Welt. Als JavaScript erfunden wurde, ging man davon aus, dass es für kurze Codeschnipsel verwendet wird, eingebettet in eine Webseite. Es ist eine Skriptsprache und wurde für die Benutzerinteraktion entwickelt. Mithilfe von JavaScript lassen sich Inhalte verändern oder neu generieren. Für das Projekt ist die Interaktion mit dem Benutzer von entscheidender Bedeutung, da er ständig neue Zahlen oder Kandidaten eingeben kann.

Diese Anforderungen lassen sich mit JavaScript umsetzten. Daher wurde sich beim Frontenddevolpment für JavaScript mit der Erweiterung von jQuery entschieden. [Ste20, S. 7 ff.] [Sta]

jQuery

Neben JavaScript wird die freie Programmbibliothek jQuery für das Frontend verwendet. jQuery bietet die Funktionalität der Document Object Model Navigation und Manipulation sowie ein erweitertes Event-System. Mit jQuery wird die JavaScript-Programmierung vereinfacht und viele Methoden können in einer einzelnen Zeile von Code umgesetzt werden. Ein weiterer Vorteil ist, dass auch jQuery einfach erweitert werden kann und sehr populär ist. [W3s]

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

2.2. Webserver

Die Studienarbeit soll zum Schluss auf einem Webserver laufen. In diesem Kapitel wird beschreiben welchen Anforderungen ein Webserver gerecht werden muss. Danach werden mit dem Apache HTTP Server und dem Nginx Webserver zwei beliebte Webserver vorgestellt und verglichen. Zuletzt wird die Designentscheidung aufgrund des Vergleichs begründet.

Ein Webserver ist in der Regel ein Server, der zur Verbreitung von Webinhalten im Inter- oder Intranet dient. Die jeweiligen Informationen und Dokumentationen können demnach weltweit oder firmenintern erreicht werden. Damit eine Website jederzeit erreichbar sein kann, muss der Webserver permanent online sein. Der Rechner, auf dem der Webserver läuft, wird als Host bezeichnet. Der Webserver ist für die zuverlässige Übertragung von statischen, wie beispielsweise von unveränderlichen HTML-Dateien, aber auch von dynamischen Dateien verantwortlich. Für dynamische Dateien muss der Webserver vor der Antwort Programmcode ausführen. Dieser Programmcode wird in diesem Fall in der Programmiersprache Python geschrieben. Mittels einer Cascading Style Sheets (CSS) Datei können die Inhalte der dynamischen Dateien weitestgehend von den Darstellungsvorgaben getrennt werden. In der HTML Datei wird folglich nur die inhaltliche Gliederung definiert und in der CSS Datei die Darstellung, wie etwa Farben oder Layout. Für die Übermittlung wird das Übertragungsprotokoll Hypertext Transport Protocol (HTTP) oder die verschlüsselte Variante Hypertext Transport Protocol Secure (HTTPS) verwendet. Ein Webserver ist in der Lage, die Inhalte auf viele verschiedene Rechner gleichzeitig zu übermitteln. Wie viele Nutzeranfragen (Requests) ein Server bearbeiten kann, hängt von der Hardware und der Auslastung des Hosts ab.

Am meisten verbreitet sind unter anderem die Webserver Apache HTTP Server und Nginx, wie in die Statistik 2.1 dargestellt. Beide Programme sind freie Software und wurden daher für das Projekt in Betracht gezogen. [Kar19] [Sta]

2.2.1. Apache HTTP Server

Der Apache Webserver wurde erstmals 1995 veröffentlicht und hat sich schnell zu einem der beliebtesten Webserver entwickelt. Er unterstützt neben Unix und Linux noch eine Vielzahl an weiteren Betriebssystemen.

Der Apache Server ist modular aufgebaut, wodurch benötigte Funktionen, die der Server nicht nativ bereitstellt, durch Module importiert werden können. Das Erstellen dynamischer

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

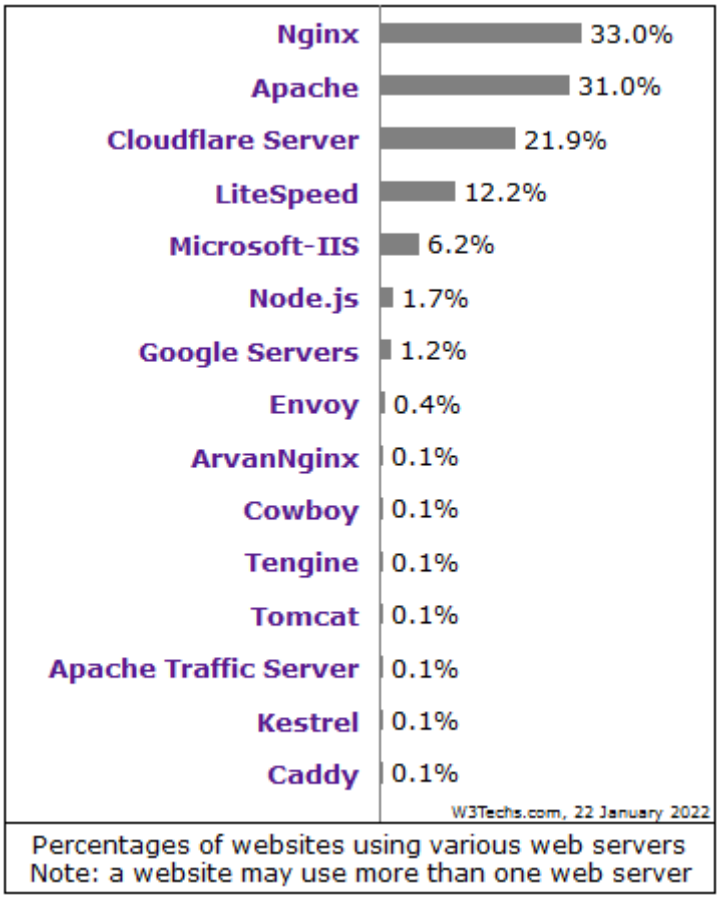

Abbildung 2.1.: Benutzungsstatistiken von Webservern [W3t]

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

Webseiten wird mittels serverseitiger Skriptsprachen bewerkstelligt. Über ein Modul kann der entsprechende Interpreter in den Server integriert werden.

Beim Apache-Webserver wird ein Ansatz verfolgt, bei dem jede Clientanfrage von einem separaten Prozess oder Thread bearbeitet wird. Dadurch werden Prozesse, die Schreib- oder Leseoperationen erfordern, nacheinander abgearbeitet und es kann passieren, dass ein Request in der Warteschlange verweilen muss, bis der vorherige Request durchgeführt werden konnte. Damit man dieses Problem umgehen kann, gibt es die Möglichkeit, mehrere Single-Threading-Prozesse gleichzeitig zu starten. Diese Strategie ist jedoch mit einem hohen Ressourcenaufwand verbunden. Um dies zu vermeiden, kommen Multi-Threading-Mechanismen zum Einsatz. Für die parallele Abfrage von Clientanfragen gibt es verschiedene Multi-Processing-Module, die integriert werden können. [Kar19] [Ion]

2.2.2. Nginx Webserver

Der Marktanteil von Nginx ist in den letzten Jahren kontinuierlich gestiegen, weshalb auch dieser Webserver für die Studienarbeit in Betracht gezogen wurde. Nginx wurde erstmals 2004 veröffentlicht und ist wie der Apache Server auch mit diversen Betriebssystemen kompatibel.

Wie Apache ist auch Nginx modular aufgebaut und verschiedene Funktionen können über Module bereitgestellt werden. Zu diesen Modulen gehört jedoch nicht die Option, Interpreter für eine Programmiersprache entsprechend in den Webserver zu integrieren. Es wird also ein weiterer Anwendungsserver benötigt. Für kleine Webprojekte ist das ein Mehraufwand, der nicht immer eingegangen werden muss.

Dieser Server zeichnet sich durch eine hohe Performance aus. Dabei können eine möglichst große Anzahl an Clients gleichzeitig bedient werden und der Ressourcenverbrauch trotzdem gering gehalten. Durch die ereignisorientierte Architektur können Client-Anfragen asynchron bearbeitet werden, wodurch Arbeitsspeicher und Zeit gespart werden kann. Die Nebenläufigkeit ist realisiert, ohne dass für jede neue Verbindung ein zusätzlicher Prozess oder Thread benötigt wird. Die Stärke dieser Architektur zeigt sich bei großen Webprojekten. [Kar19] [Ion]

2.2.3. Fazit

Die folgende Tabelle 2.1 zeigt die Unterschiede und Gemeinsamkeiten der beiden Webserver auf.

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

Merkmal	Apache	NGINX
Funktion	Webserver, Proxy-Server	Webserver, Proxy-Server, E-Mail-Proxy, Load-Balancer
Betriebssystem	Alle unixoiden Plattformen, Windows	FreeBSD, Linux, Solaris, IBM AIX, HP-UX, macOS, Windows
Lizenz	Apache License v2.0	BSD-Lizenz
Entwickler	Apache Software Foundation	Nginx, Inc.
Statische Webinhalte	Ja	Ja
Dynamische Webinhalte	Ja	Nein
Software-Architektur	Prozess-/threadbasiert	Eventgesteuert

Tabelle 2.1.: Vergleich der Webserver Apache HTTP Server und Nginx Webserver

Der Apache Webserver bietet eine breite Möglichkeit an Modulen, um die Software zu erweitern. Der ausschlaggebende Grund, warum sich in diesem Projekt für den Apache HTTP Server entschieden wurde, ist die Möglichkeit Interpreter für Programmiersprachen über ein Modul direkt in den Webserver zu integrieren. Zudem muss für das Projekt nur ein Client bedient werden. Dadurch wird für das Anzeigen von dynamischen Inhalten kein separaten Anwendungsserver benötigt. Damit bietet der Apache HTTP Server eine bequemere Lösung für kleine Websites, deren Inhalt dynamisch erzeugt wird.

2.3. Framework

Dieser Abschnitt stellt zwei verschiedene Frameworks vor, die für die Studienarbeit in Frage kommen. Auch hier wird zunächst erläutert was ein Framework ist und anschließend zwei vorgestellt, verglichen und die Nutzungsentscheidung darauf basierend begründet.

Ein Framework ist ein Programmiergerüst. Es wird insbesondere bei komponentenbasierten und bei der objektorientierten Softwareentwicklung verwendet. Ein Framework ist kein fertiges Programm, sondern stellt den Rahmen, in der eine Anwendung erstellt werden soll, zur Verfügung. Frameworks lassen sich in Typen gliedern, wobei es nicht immer eine strikte Trennung gibt. Zu diesen Typen gehören beispielsweise Application Frameworks oder Webframeworks.

Die beiden Frameworks, Django und Flask, die für das Projekt in Betracht gezogen wurden, gehören zu den beliebtesten Webframeworks für Python und sind für die Entwicklung von dynamischen Webanwendungen ausgelegt. [Ste20, S. 195 ff.] [Sta] [Lub22]

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

2.3.1. Django

Django ist ein kostenloses Open Source Webframework und kann plattformübergreifend genutzt werden. Es folgt dem MVC Entwurfsmuster, das auch in der Studienarbeit genutzt wird. Der Fokus von Django liegt in einer schnellen Entwicklung mit weniger Code. Mit dem Framework kommen viele bereits eingebaute Pakete, die genutzt werden können. Dazu gehört unter anderem die integrierte Anbindungen an verschiedenen Datenbanksysteme. Django hat verschiedene Mechanismen integriert, um die Sicherheit der Website zu garantieren. So können SQL Injektionen oder Cross-site Scripting ausgeschlossen werden.

Ein weiterer Vorteil von Django ist die große und aktive Community, wodurch viel Fragen schnell beantwortet werden können und es gibt es eine ausführliche Dokumentation zu dem Framework. [Her22] [Lub22]

2.3.2. Flask

Webseiten, die Flask als Framework nutzen, sind meistens Single-Page-Webanwendungen. Flask ist im Gegensatz zu Django modular und eher minimalistisch aufgebaut. Daher muss in der Entwicklung tendenziell mit externen Bibliotheken gearbeitet werden. Man nennt Flask deshalb auch ein Mikroframework. Für die Sicherheit der Website gibt es die Flask-Security Bibliothek, die dieselben Mechanismen wie Django beinhaltet.

Aus diesen Gründen ist Flask eher für kleinere Projekte geeignet, die benutzerdefinierte Komponenten erfordern oder für das Prototyping. Die Anzahl der Webseiten, die Flask als Framework nutzen, steigt stetig weiter. Der Vorteil, warum Flask immer beliebter wird ist die Kontrolle, die man über das Projekt bekommt, indem man die Komponenten selbst bestimmen kann. [Her22]

2.3.3. Fazit

Beide Frameworks wären für das Projekt geeignet. Die Django Community ist größer und aktiver als die von Flask. Daher gibt es mehr Informationen über Django. Jedoch wird Flask mittlerweile in mehreren Projekten genutzt. In der Tabelle 2.2 sind die wichtigsten Eigenschaften nochmals gegenübergestellt.

Aufgrund der Leichtigkeit wurde sich bei den Frameworks für Flask entschieden. Da es bei der Performance keine großen Unterschiede zwischen Django und Flask gibt, ist es als Merkmal für die Entscheidung eines Frameworks nicht relevant. Für den **Sudoku Helper**

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

Merkmal	Django	Flask
Use cases	mittelgroße bis große Projekte	kleine bis mittelgroße Projekte
Community	große, aktive Community	kleinere Community
Packages	built-in Packages	modular
Performance	gut	gut

Tabelle 2.2.: Vergleich der Frameworks Django und Flask

wird nur eine einseitige Anwendung ohne Datenbankanbindung benötigt, wodurch Flask das geeignetere Framework ist. Auch der modulare Aufbau spricht für Flask, da nur wenige zusätzliche Pakete genutzt werden.

Das Entwurfsmuster MVC wird im Framework Flask übernommen.

2.4. Entwurfsmuster

In der Studienarbeit wird für die Übersichtlichkeit ein Entwurfsmuster verwendet. Der folgende Abschnitt stellt das Entwurfsmuster und die Komponenten vor.

Wie im Abschnitt der Frameworks schon erwähnt, wird in diesem Projekt eine bestimmte Softwarearchitektur verwendet, damit der Programmcode eine gewisse Übersichtlichkeit behält. Damit werden in einem Projekt die grundlegenden Komponenten und das Zusammenspiel dieser visualisiert. Damit ist es einfacher nicht-funktionale Eigenschaften wie die Modifizierbarkeit, Wartbarkeit oder Sicherheit darzustellen und zu verstehen. [Mus21, S. 3 f.]

Das Framework Django arbeitet immer mit dem MVC Architekturmuster. Da aber Flask als Framework genutzt wird, muss diese Architektur selbst implementiert werden. Im Folgenden wird das Architekturmuster mit seinen drei Komponenten kurz beschrieben.

2.4.1. Model View Controller

Das Ziel dieser Entwurfsarchitektur ist es, spätere Änderungen oder Erweiterungen zu vereinfachen und die Wiederverwendbarkeit der einzelnen Komponenten zu steigern. In der Abbildung 2.2 ist die Architektur visualisiert. Durchgezogene Linien stehen hierbei für eine direkte Assoziation und gestichelte für eine indirekte Assoziation. [Mus21, S. 57 f.]

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

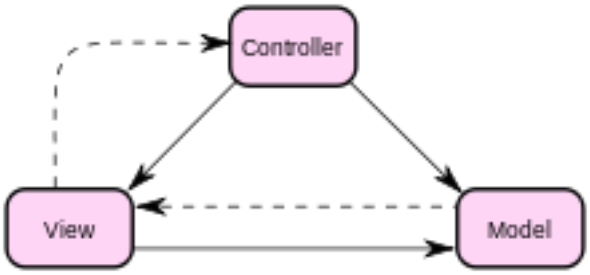

Abbildung 2.2.: MVC Konzept

Model

Im Model sind die Daten enthalten, die dargestellt werden sollen. In Falle des Sudoku Solvers ist hier das Sudoku Board gespeichert.

View

In der View oder der Präsentation werden die Daten aus dem Model dargestellt. Außerdem wird in der View die Benutzerinteraktion geregelt. Die View kennt die Daten aus dem Model, soll diese aber nicht bearbeiten oder verändern und ist völlig unabhängig von dem Controller.

Controller

Im Controller werden das Model und die View verwaltet. Die View unterrichtet den Controller von Benutzerinteraktionen. Im Controller wird diese dann ausgewertet und gegebenenfalls das Model und die View angepasst. In dieser Komponente sind die Lösungsstrategien für den Sudoku Solver implementiert.

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

3. Sudoku

Nachdem in der Einleitung schon kurz auf das Spielfeld eines Sudokus und die Regeln eingegangen wurde, werden beides in diesem Kapitel noch einmal ausführlicher behandelt. Es wird auch auf Elemente des Spielfelds eingegangen, die relevant sind für die Strategien und wie das Spielfeld in dem Programm aufgebaut ist und eine einheitliche Identifikation definiert. Im letzten Abschnitt wird das Prinzip der Strategien, um ein Sudoku lösen zu können, behandelt. Da für die Strategien und den **Sudoku Solver** die Lösungshilfe der Kandidaten-Notation eine wichtige Rolle spielt, wird darauf in einem weiteren Unterkapitel eingegangen und die Funktion und Bedeutung erklärt.

3.1. Spielfeld Aufteilung

Zunächst wird die Aufteilung des Spielfelds erläutert und die Units vorgestellt. Jede Unit bekommt einen eindeutigen Identitiver, auf den im weiteren Verlauf der Arbeit referenziert werden kann.

Das Spielfeld, auf dem ein Sudoku gespielt wird, besteht aus einem 9x9 Gitter und damit aus 81 einzelnen Zellen. Diese Gitter wird weiterhin in je neun Zeilen, Spalten und 3x3 Blöcke aufgeteilt. Jede Aufteilung, weiterhin auch Unit genannt, besteht immer aus neun Zellen. Damit kann ein Sudoku also nochmal in 27 verschiedene Units aufgeteilt werden. Wie diese Units aussehen, wird im Folgenden gezeigt. [Sud22]

3.1.1. Zeile

Eine Zeile in einem Sudokuspielfeld sind Zellen, die horizontal zueinander ausgerichtet sind. In der Abbildung 3.1 wurde die oberste Zeile beispielhaft markiert. Auf die Zeilen wird ab jetzt und im Folgenden der Arbeit mit den Zahlen 1 bis 9 referiert.

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

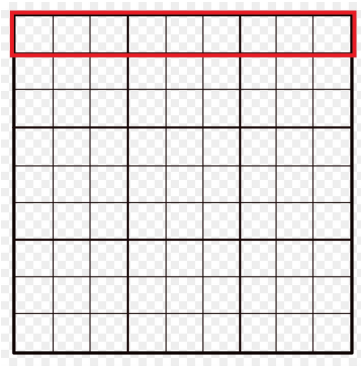

Abbildung 3.1.: Darstellung des Sudokugitters mit der Markierung einer Zeile

3.1.2. Spalte

Analog zu einer Reihe sind Spalten Zellen, die vertikal zueinander ausgerichtet sind. In der Abbildung 3.2 wurde beispielhaft die erste Spalte markiert. Die Spalten werden ähnlich zu

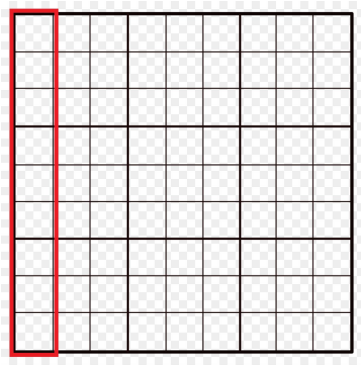

Abbildung 3.2.: Darstellung des Sudokugitters mit der Markierung einer Spalte

den Zeilen von links nach rechts nicht mit Zahlen, sondern mit den Buchstaben A bis I beschrieben.

3.1.3. Box/Block

Boxen bestehen aus 3x3 Zellen. Die 3x3 Boxen sind im Sudokufeld so angeordnet, dass jede Zelle in genau einer Box liegt und, dass es neun Boxen geben kann. In der Abbildung 3.3 wurde eine Box markiert. Wie zu erkennen ist, sind die Boxen auch im Frontend durch dickere Linien voneinander abgegrenzt. Blöcke werden durch römische Zahlen identifiziert. Dabei wird erst von links nach rechts und von dann von oben nach unten mit I bis IX nummeriert.

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

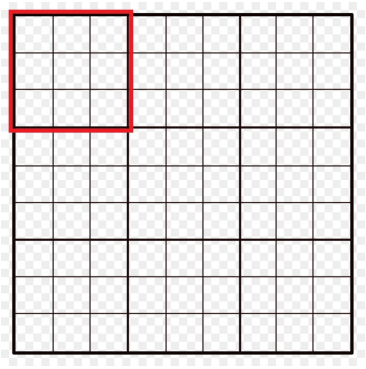

Abbildung 3.3.: Darstellung des Sudokugitters mit der Aufteilung in 9 Boxen

3.2. Regeln

Nachdem die Abgrenzungen der Units und damit die Aufteilung des Spielfelds definiert wurde, werden die Regeln auf Basis des jetzigen Wissensstand nochmals genauer erläutert.

Die Regeln ergeben sich durch einfache Bedingungen, wenn man die Aufteilung des Spielfeldes kennt. In jeder Zeile, Spalte und Box dürfen und müssen die Zahlen von eins bis neun nur einmal vorkommen.

Am Ende kommen die Zahlen von eins bis neun also jeweils neunmal auf dem gesamten Sudokugitter vor, ohne dass sie sich gegenseitig sehen, also sich keine Zahl doppelt in einer Zeile, Spalte oder Box befindet.

Damit man das Sudokugitter ausfüllen kann und es nur eine mögliche Lösung gibt, sind zu Spielbeginn einige Ziffern vorgegeben. Mithilfe dieser Ziffern ist es möglich, mit den gerade erklärten Regeln das ganze Sudokugitter durch Logik und Deduktionen zu füllen. [Sud22] [Mar]

3.3. Strategie

In diesem Abschnitt wird die Bedeutung von Strategien im Sinne des Lösens eines Sudokus beschrieben. Es werden zwei verschiedene Arten von Strategien angesprochen. In diesem Sinne wird das Hilfsmittel der Kandidaten-Notation gebraucht und daher erläutert.

Um ein Sudoku zu lösen, können verschiedene Strategien eingesetzt werden. Das Ziel einer Strategie ist es hierbei eine passende Zahl in das Sudoku einzufügen oder mögliche Kandidaten zu eliminieren. Es gibt zwei grundlegende Strategien, um Zahlen direkt einzufügen: den Hidden Single und den Naked Single. Beide Strategien können oft intuitiv

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

und ohne eine große Erklärung gefunden und umgesetzt werden. Weitere leichte Strategien wie ein Hidden Double oder ein Naked Double, bei denen Kandidaten eliminiert werden können, sind mithilfe der Kandidaten-Notation leicht gefunden.

Insgesamt gibt es über 30 gebräuchliche Strategien, die zur Unterstützung bei schweren Sudokus angewendet werden können. Oftmals sind anspruchsvollere Strategien für das menschliche Auge schwer zu entdecken. Wichtig bei diesen komplexeren Strategien ist eine gewissenhafte Ausführung der Kandidaten-Notation. [Mar] [Zam15, S. 9]

3.3.1. Lösungshilfen: Kandidaten-Notation

Kandidaten sind Zahlen in einer Zelle des Sudokugitters, die, mit dem aktuellen Informationsstand, als feste Zahl noch infrage kommen. Das Problem liegt dabei, dass meistens mehrere Zahlen in eine Zelle können. Kandidaten können durch das Eintragen von weiteren Zahlen oder das Anwenden von Strategien eliminiert werden. Wenn man dabei jede Zelle einzeln betrachtet, geben die Kandidaten keinen weiteren Nutzen. Wenn die Kandidaten jedoch über mehrerer Zellen in Bezug zueinander angeschaut werden, lassen sich Zusammenhänge erkennen. Die meisten Strategien basieren auf diesen Zusammenhängen.

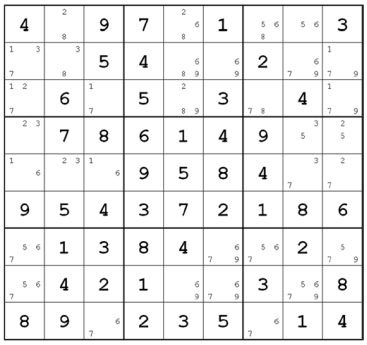

Abbildung 3.4.: Beispiel Rätsel mit Kandidatennotation

Sobald eine Zahl eingetragen wird, müssen die Kandidaten aktualisiert werden. Wenn eine Zahl eingetragen wird, kann jeder gleichwertiger Kandidat, der diese Zahl sieht, gestrichen werden. [AV14, S. 85 f.] [Mar] [Zam15, S. 4]

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

4. Die Mathematik hinter Sudoku

In diesem Kapitel sollen einige mathematische Fragen über Sudokus geklärt werden. Dazu gehören Fragen zur eindeutigen Lösbarkeit und wie diese bewiesen werden kann oder wie viele mögliche fertige Sudokus es denn überhaupt gibt. Zudem wird eine mathematische Beschreibung eines Sudokus gegeben und eine algorithmische Lösungsmethode, die ohne das Anwenden von Strategien, eine Lösung für ein Sudoku findet, falls es eine Lösung gibt.

4.1. Abzählfragen

In diesem Abschnitt soll geklärt werden wie viele mögliche Lösungen für ein vollständig ausgefülltes Sudokugitter existieren. Dabei werden zwei verschiedene Ansätze beschrieben. Besonders der Ansatz von Jarvis und Felgenhauer wird genauer angeschaut.

Ein trivialer Ansatz, um die Frage zu beantworten, wie viele vollständig ausgefüllten 9×9 Standard-Sudokus es gibt, könnte sein zeilenweise von links nach rechts oder spaltenweise von oben nach unten Zahlen einzutragen. Damit würden sich pro Zeile, Spalte oder Block 9! Möglichkeiten ergeben. Daraus ergeben sich

$$(9!)^9 = 1{,}1 \cdot 10^{50} \tag{4.1}$$

Möglichkeiten, ein Sudokugitter auszufüllen. Von diesen ausgefüllten Gittern entsprechen aber nicht alle der Konvention, dass in den Zeilen, Reihen und Boxen jede Zahl von 1 bis 9 nur einmal vorkommen darf.

Felgenhauer und Jarvis hatten für eine einzelne Unit dieselbe Herangehensweise. Anschließend haben Sie die möglichen Lösungen von drei linear verbundenen Blöcken errechnet, wie in Abbildung 4.1 dargestellt. Für die Erklärungen werden im weiteren Verlauf exemplarisch die Blöcke I, II und III verwendet.

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

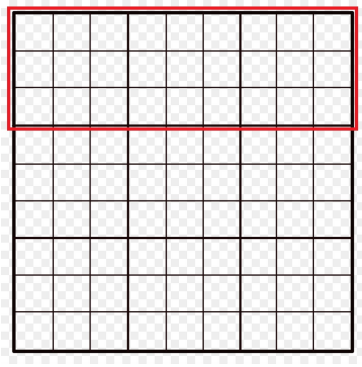

Abbildung 4.1.: Markierung von drei linear verbundenen Blöcken

Als Ausgangspunkt setzten Felgenhauer und Jarvis die Box I als ausgefüllt voraus. Zunächst berechnen sie die Anzahl von Möglichkeiten den Rest der obere Reihe mit zwei Sets aus jeweils drei Zahlen auszufüllen. So ein Dreier Set in das die Zahlen eingetragen werden sollen ist in Abbildung 4.2 markiert. Jedes Set aus drei Zahlen kann in $3! = 6$ verschiedenen Anordnungen in die oberste Zeile eingetragen werden. Dabei müssen alle sechs Sets, die die Blöcke II und III formieren mit in Betracht gezogen werden. Dasselbe gilt für die Vertauschung der Anordnung der Sets der obersten Reihe. Dadurch ist der erste Teil $2 \cdot (3!)^6$ der folgenden Formel erklärt.

Alle anderen 18 Möglichkeiten zwei dieser dreier Sets auszufüllen verhalten sich anders. Nachdem die obere Reihe in Block II und III ausgefüllt wurde, können Zahlen in den übrigen vier Sets aufgrund von fehlenden Einschränkungen vertauscht werden. Damit haben Felgenhauer und Jarvis für den zweiten Teil der Gleichung, $18 \cdot 3 \cdot (3!)^6$ argumentiert. Insgesamt gibt es also 56 Konfigurationen die sechs Dreier Sets weiter auszufüllen.

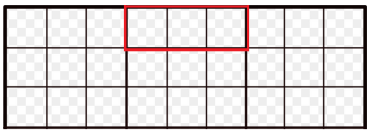

Abbildung 4.2.: Markierung eines Dreier Sets als Kombination aus Block und Zeile

Damit gibt es

$$2 \cdot (3!)^6 + 18 \cdot 3 \cdot (3!)^6 = 56 \cdot (3!)^6 = 2612736 \tag{4.2}$$

mögliche Erweiterungen einer bereits ausgefüllten Unit zu drei linear verbundenen Blöcken.

Da es für eine vollständig ausgefüllte Unit aber 9! unterschiedliche Ausführungen gibt, muss man die die Möglichkeiten eine Unit auszufüllen mit den möglichen Ergänzungen multiplizieren.

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

Damit gibt es

$$9! \cdot 2612736 = 948109639680 \tag{4.3}$$

Möglichkeiten, die oberen drei Reihen eines Sudokugitters auszufüllen.

Aus diesen 2612736 Möglichkeiten wollten Felgenhauer und Jarvis dann versuchen, die verbleibenden Blöcke des Sudokugitters mittels einer Brute-Force Methode auszufüllen. Da es mit den 2612736 Möglichkeiten immer noch ein zu großer Aufwand wäre, haben sich Felgenhauer und Jarvis drei Reduktionen überlegt, um die Anzahl an Möglichkeiten die ersten drei Reihen auszufüllen zu reduzieren.

Auf diese 2612736 Möglichkeiten haben Jarvis und Felgenhauer drei möglichen Reduktionen angewandt: die lexikographische Reduktion, die Reduktion der Permutationen und die Reduktion der Spalten. Durch diese Reduktionen werden Sudokus mit Lösungen, die sich durch Spiegeln, Drehen oder durch symmetrische Eigenschaften doppeln lassen, herausgefiltert. Aus Praktikabilitätsgründen kann in dieser Arbeit keine umfassende Übersicht der Reduktionen gegeben werden.

Nach den Reduktionen bleiben 44 Möglichkeiten, die Zeilen zwei und drei auszufüllen. Danach muss herausgefunden werden, wie viele Konfigurationen eines Sudokus es gibt, das Gitter zu vervollständigen.

Felgenhauer und Jarvis haben in ihrer Veröffentlichung von 2006 die These aufgestellt, dass es 6.670.903.752.021.072.936.960, also ca. 6,7 Trilliarden verschiedene, vollständig ausgefüllte Standard-Sudokus Rätsel gibt. Nach der Reduktion von symmetrischen Lösungen bleiben nur noch 5.5 Milliarden Lösungen. [FJ06] [AV14, S. 90 ff.]

4.2. Komplexität

Das folgende Unterkapitel gibt eine mathematische Definition eines Sudoku. Dabei wird eine allgemeine Beschreibung gegeben, die auf Sudokus die ein Problem n-ter Ordnung sind, angewendet werden kann.

Bei Sudokus, die in den Medien veröffentlicht werden, und immer eindeutig lösbar sein müssen, kann durch logische Schlussfolgerung das Sudokurätsel gelöst werden. Durch die eindeutige Lösbarkeit ist es nicht notwendig Fallunterscheidungen zu treffen, also auszuprobieren.

Mathematisch gesehen ist ein Sudoku ein Problem n-ter Ordnung. Dabei ist n eine natürliche Zahl. Auf einem $N \cdot N$ Gitter, wobei $N = n^2$ gilt, sind die Zahlen 1 bis N in

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

das Gitter einzufügen. Dabei soll in jeder $1 \cdot n$ Zeile, $n \cdot 1$ Spalte und $n \cdot n$ Block, jede der Zahlen 1 bis N genau einmal vorkommen. N^2 Felder des Gitters können dabei vorbelegt sein. Ein $9 \cdot 9$ Sudoku ist damit ein Problem 3-ter Ordnung. [HM07]

4.3. Eindeutige Lösbarkeit

Eine wichtige Eigenschaft von Sudokurätseln ist die eindeutige Lösbarkeit. In diesem Abschnitt wird erklärt was genau unter eindeutiger Lösbarkeit gemeint ist und wie viele Ziffern vorgegeben sein müssen um das zu erreichen.

Für ein Sudokugitter ohne vorgegebene Ziffern gibt es 5.472.730.538 (5,5 Milliarden) unterschiedliche, richtige Lösungen. Auch wenn nur eine oder zwei Ziffern vorgegeben werden, gibt es immer noch eine sehr große Anzahl an Lösungen für dieses Sudoku.

Sudokus, die in irgendeiner Form veröffentlicht werden, sind normalerweise mit der Vorgabe einer eindeutigen Lösung erstellt. Sobald ein Sudoku nur eine korrekte Vervollständigung hat, ist es eindeutig lösbar. Daraus lässt sich folgern, dass in eindeutigen Sudokus in jede freie Zelle nur eine einzige Ziffer eingetragen werden kann, ohne die Regeln zu brechen.

Als notwendiges Kriterium für ein eindeutig lösbares Sudoku, müssen unter den vorgegebenen Zahlen mindestens acht unterschiedliche Zahlen von 1 bis 9 vorkommen. Dieses Kriterium ist gegeben durch den Fakt, dass bei nur sieben vorgegebenen unterschiedlichen Ziffern die beiden übrigen in der zugehörigen Lösung vertauscht werden können. [HM07] [AV14, S. 95 ff.]

4.3.1. Anzahl vorgegebene Ziffern für ein eindeutiges Sudoku

Es gibt die Vermutung, dass die minimale Anzahl an vorgegebenen Ziffern 17 ist. Mittels Brute-Force Methoden wurden eindeutige Sudokus mit nur 17 vorgegebenen Zahlen gesucht. Daran wurde 2011 von einem Forschungsteam von Gary McGuire geforscht. Es gibt jedoch noch keinen mathematischen Beweis für die Vermutung. Diese Vermutung basiert also hauptsächlich auf dem Generieren und Ausprobieren von vielen unterschiedlichen Sudokurätseln mit nur 17 Ziffern und einer eindeutigen Lösung. [Zei12] [AV14, S. 95 ff.]

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

4.4. Algorithmische Lösungsmethode: Backtracking

Um die eindeutige Lösbarkeit beweisen zu können muss ein Sudokurätsel gelöst werden. In der Studienarbeit wird das Überprüfen der eindeutigen Lösbarkeit mittels der algorithmischen Lösungsmethode des Backtracking bewiesen. Dieser Algorithmus wird im Verlaufe des Abschnitts erklärt und wie damit der Beweis der eindeutigen Lösbarkeit umgesetzt werden kann.

Backtracking ist eine Problemlösungsstrategie und mit der Rekursion verwandt. Es werden alle möglichen Lösungen ausprobiert und in jedem Schritt nach einer Abbruchbedingung überprüft. Die Bedingungen an das Problem, dass mit Backtracking gelöst werden sollen sind die Folgenden:

1. Ein Problem kann in endlich viele Teilschritte aufgeteilt werden
2. Ein Teilschritt besitzt eine Abbruchbedingung
3. Ein Teilschritt hat endlich viele Lösungsmöglichkeiten

Für jedes Element des Problems, in diesem Fall für jede freie Zelle, werden alle möglichen Zustände ausprobiert. Die Zustände sind im Fall des Sudokus die Zahlen von eins bis neun. Wenn ein Zustand zulässig ist, also in der Zeile, Reihe oder Box die Zahl nicht bereits eingetragen ist, so wird rekursiv überprüft, ob es für den aktuellen Zustand eine Lösung gibt. Wenn es diese nicht gibt, wird der vorherige Schritt rückgängig gemacht und eine neue Lösung gesucht. Damit basiert Backtracking auf dem Trial-and-Error Prinzip und versucht eine erreichte Teillösung in eine Gesamtlösung zu transferieren.

Bei z möglichen Verzweigungen jeder Teillösung und einem möglichen Verzweigungsbaum mit der Tiefe von N hat das Backtracking, sofern $z > 1$ ist im schlechtesten Fall mit $O(z^N)$ eine exponentielle Laufzeit.

$$1 + z + z^2 + z^3 + ... + z^N \tag{4.4}$$

Wenn mittels des Backtrackings keine Lösung für das Sudoku gefunden wurde, gibt es keine Lösung. Wenn jedoch eine Lösung gefunden wurde, so ist noch nicht bewiesen, dass diese Lösung eindeutig ist. Um eine eindeutige Lösbarkeit zu garantieren, wird mittels Backtracking überprüft, dass keine weitere Lösung existiert.

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

Da es für das Lösen eines Sudokus keinen effizienteren Algorithmus gibt und bereits vor der Anwendung der Lösungsstrategie überprüft werden soll, ob es eine eindeutige Lösbarkeit gibt, ist dieser Ansatz am sinnvollsten. [Log14, S. 209 ff.] [Kno17]

4.4.1. Backtracking mit Brute-Force-Methode

Backtracking mit der Brute-Force Methode funktioniert wie oben beschrieben auf dem Prinzip des Backtracking mit dem Ausprobieren aller möglichen Fällen. Mit dem ersten freien Feld probiert man mit der 1 beginnend, ob man zu einer Lösung kommt. Als Abbruchbedingung ist das nicht singuläre Auftreten der neu eingefügten Zahl in Reihe, Zeile und Box implementiert.

Bei der Brute-Force Methode wird für jede freie Zelle von 1 bis 9 überprüft, ob diese Zahl in die Zelle eingetragen werden darf. Sobald eine Zahl eingetragen werden darf, wird das gemacht und mit der nächsten freien Zelle fortgefahren. Falls in einer Zelle die Abbruchbedingung für jede Zahl ausgelöst wurde, also keine Zahl mehr in die Zelle eingetragen werden kann, wird bei der vorherigen Zelle weiter versucht. Bis das Sudoku entweder vollständig gelöst ist, oder es keine Möglichkeiten mehr gibt und das Sudoku damit nicht lösbar ist.

Die Laufzeit des Algorithmus hängt von der Anzahl der freien Zellen ab und damit auch von der Anzahl der vorgegebenen Zahlen. Wenn viele Zahlen vorgegeben sind, dann ist die Tiefe N des Verzweigungsbaums und damit auch die Laufzeit geringer.

```
1   def solve(self, numbers=(1, SIZE + 1), board=None):
2     """
3     Solves the SudokuBoard recursively via backtracking
4     :return: False if not solvable, True if solved
5     """
6     square = self.get_empty_square(board)
7
8     if not square:
9       return True
10
11    for i in range(*numbers):
12      if self.is_valid(i, square, board=board):
13        board[square[0]][square[1]][0] = i
14
15        if self.solve(numbers=numbers, board=board):
16        return True
17
```

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

```
18          board[square[0]][square[1]][0] = 0
19
20      return False
```

Listing 4.1: Funktion zum Lösen des Sudokurätsel

Abbruchbedingung

Die Abbruchbedingung für das Backtracking ist mit einer eigenen Funktion *is_valid* umgesetzt.

```
1   def is_valid(self, number, position, board=None):
2       ...
3       # Check row
4       for j in range(SIZE):
5           if board[position[0]][j][0] == number and j != position[1]:
6           return False
7       ...
8       # Check Box
9       box_x = (position[1] // 3) * 3
10      box_y = (position[0] // 3) * 3
11
12      for i in range(box_y, box_y + 3):
13          for j in range(box_x, box_x + 3):
14              if board[i][j][0] == number and (i, j) != position:
15              return False
16
17  return True
```

Listing 4.2: Abbruchbedingung durch Validierung der Lösungsvariante

Wie in dem Code implementiert, wird überprüftm, ob eine *number* an einer bestimmten *position* eingesetzt werden darf, oder ob diese Zahl bereits in der entsprechenden Reihe, Zeile oder Box vorhanden ist. Wenn die Zahl gültig ist, gibt die Funktion *True* zurück, anderenfalls *False*, womit die Abbruchbedingung als erfüllt gilt. [Kno17]

4.4.2. Beweis Eindeutigkeit

Der Algorithmus des Backtrackings bricht nach dem Finden einer vollständigen Lösung für ein Sudoku ab. Damit ist bestätigt, dass es eine Lösung für das Sudoku Rätsel gibt, jedoch

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

nicht, dass diese Lösung eindeutig ist. Um die Eindeutigkeit eines Sudokus zu beweisen, müsste nach dem Finden einer Lösung erfolglos versucht werden eine weitere Lösung zu finden.

Um die Eindeutigkeit eines Sudoku Rätsels zu beweisen, wird nach dem Finden einer Lösung das Backtracking nochmals von hinten begonnen. Damit ist gemeint, dass bei der ersten Lösungssuche in den freien Zellen die Zahlen von 1 bis 9 eingetragen werden und nach dem Abbruchkriterium überprüft wird. Beim Suchen nach einer zweiten Lösung beziehungsweise bei dem Beweis der Eindeutigkeit wird das Backtracking nochmals ausgeführt, dabei aber die Zahlen von 9 bis 1 in die freien Zellen eingetragen.

```
1  def is_uniquely_solvable(self):
2    temp_board = deepcopy(self.board)
3    self.solve(numbers=(9, 0, -1), board=temp_board)
4    return self.solved == temp_board
```

Listing 4.3: Überprüfen ob ein Sudokurätsel eine eindeutige Lösung hat

Nach dem Ausführen der beiden Backtracking Algorithmen werden die beiden gefundenen Lösungen miteinander verglichen. Wenn die dabei gefundenen Lösungen übereinstimmen, kann mit Garantie gesagt werden, dass das Sudoku eindeutig lösbar ist.

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

Teil II.

Implementierung

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

5. Frontend

Nachdem in den bisherigen Kapiteln eher die Designentscheidungen und die theoretischen Grundlagen diskutiert wurden, wird nun die eigentliche Implementierung des **Sudoku Solver** erläutert.

Zunächst wird das Frontend mit allen Funktionalitäten zum UI, UX und der Umsetzung der einzelnen Hilfestellungen beschrieben.

Das Frontend ist letztlich alles was der User auf der Website sehen kann. Alles was der Nutzer nicht sehen kann gehört zum Backend. Das Frontend ist trotzdem sehr entscheidend, denn es sollte möglichst Benutzerfreundlich sein. Dazu gehört eine intuitive Nutzung der Oberfläche und im Falle des **Sudoku Solver** das übersichtliche Anzeigen der Lösungsstrategie und die Erklärung dazu, sowie die sprechende Markierung auf dem Sudokuboard.

5.1. User Interface

Das UI ist die Nutzerschnittstelle zum Backend. An dieser Stelle beeinflusst der Nutzer durch seine Handlung die Reaktion im Backend und wie das Frontend letztlich wieder auf diese Handlung reagiert. An dieser Stelle der Ausarbeitung werden die Designentscheidungen für das User Interface (UI) vorgestellt. In diesem Abschnitt wird daher die HTML Struktur und das darauf basierende responsiv Design beschrieben.

Ein wichtiger Teil des UI ist das dem Nutzer die Funktionalitäten übersichtlich und klar zur Verfügung gestellt werden. Wenn das UI schon unübersichtlich und schlecht zu verstehen ist, dann wird auch die UX keine gute sein, egal wie gut die Software im Backend letztlich ist. Eine weitere Anforderung an das Frontend ist ein responsiv Design. Die Website soll möglichst sich möglichst allen Formaten anpassen und auf verschiedenen Endgeräten ihren Nutzen erfüllen.

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

5.1.1. HTML Struktur

DIe HTML Datei ist unterteilt in *<head>* und *<body>*.

Im Head werden Informationen über die Seite, wie beispielsweise der Titel oder die Stylesheets. Ebenso werden die benutzten Skripts aus den externen Dateien in das Dokument eingebunden.

Im Body sind die darzustellenden Inhalte eines HTML Dokuments enthalten und weil das Produkt hauptsächlich auf einem Computer genutzt wird, ist die HTML Struktur im Body darauf angepasst.

TODO: Abbildung Frontend

In der Abbildung xx ist das Frontend zu sehen. Das Sudokuboard, das NumPad und die Erläuterung der Strategie liegen jeweils nebeneinander. Das Sudokuboard wird rechts und in einer zur Fenstergröße passenden Höhe abgebildet, damit man alle Zahlen gut sehen kann und direkt sieht worum es geht. Rechts daneben wird das NumPad platziert. Dadurch ist durch den Platzhalter für das Anzeigen der Strategie mit Erläuterung keine Lücke zwischen den Objekten, die für eine Nutzerinteraktion relevant sind. Um das NumPad nicht zu verschieben wenn der Titel einer Strategie angezeigt wird, hat das Feld für Titel und Erklärung einen Platzhalter links neben dem NumPad.

Die Lösung ist dabei so aufgebaut, dass das Sudokuboard in einem Div liegt und das NumPad und das Textfeld für Titel und Erklärung sind ein weiteres Div teilen. Dieses Div von NumPad und Textfeld ist jeweils nochmal in zwei Divs unterteilt. In der folgenden Abbildung xx ist die Unterteilung des Elemente nochmals visualisiert.

TODO: Bild mit roten Kästen um jeweilige Divs

5.1.2. Responsiv Design

Die in der HTML Struktur beschriebenen Designentscheidungen wurden hauptsächlich von dem Responsiv Design beeinflusst.

Den Elementen wird die Layoutmethode der Flexbox gegeben. Damit werden die Elemente automatisch gestreckt, wenn es mehr Platz zum Füllen gibt und sie werden gestaucht, wenn es weniger Platz gibt. Mit den Flexboxen ist es daher relativ einfach möglich ein responsiv Design zu schaffen.

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

Wenn das Fenster kleiner wird, dann skaliert sowohl das Board als auch das NumPad und das Textfeld mit. Falls das Fenster zu schmal wird, sodass Board und Kombination aus NumPad und Textfeld nebeneinander kein Platz haben, dann wird das Element von NumPad und Textfeld unter dem Board angezeigt.

Insgesamt wird das Board so skaliert, dass der Nutzer in der horizontalen nicht scrollen muss, um das gesamte Board zu sehen. Das ist aber nur bis zu einer gewissen Breite sinnvoll, da sonst das Board zu klein wird. Diese Breite wurde basierend auf dem Frontend eines mobilen Endgeräts bestimmt.

In der Flexbox von NumPad und Textfeld sind beide Divs jeweils nochmal eigene Flexboxen. Im anderen Fall ergab sich das Problem, dass die Position des NumPad abhängig von welche Stufe der Hilfestellung immer verändert war. Das ergab keine gute User Experience (UX). Durch diese beiden Flexboxen ist auch das Anzeigen bei schmaleren Fenstern einfacher geworden. In diesem Fall wird das NumPad und die Erläuterung nebeneinander unter dem Board angezeigt wie in Abbildung xx dargestellt. Die Flexbox von von NumPad und Textfeld ist dabei genauso breit wie das Board selbst.

TODO: Bild wie aussieht wenn untereinander

Ein weiterer Punkt der beim responsiv Design beachtet werden muss, ist die Schriftgröße. Eine Anpassung wird deshalb bei den Zahlen und Kandidaten auf dem Board und auch bei der Größe der Tasten des NumPad und die Schrift darauf vorgenommen. Das Gleiche gilt für das Textfeld mit Lösungsstrategie und Erläuterung. Die Skalierung der Schriftgröße wird in der CSS Datei über viewports designt. Damit die Schrift am Ende nicht zu klein ist, wird eine *min-font-size* definiert.

5.2. User Experience

Der folgende Abschnitt ist eine Ausführung über verschiedene Funktionalitäten, die für eine guten User Experience implementiert sind. Dabei wird unterschieden zwischen den Funktionalitäten vom NumPad und anderen Eingaben und den Funktionalitäten des Sudokuboards.

Neben einer übersichtlichen und verständlich aufgebauten UI gehört unter anderem noch eine gute UX. Dazu gehören die Funktionalitäten der Knöpfe und eine angenehme Auswahl an Farben zur Markierung, um dem User die Nutzung der Website möglichst angenehm zu gestalten. Für das Eintragen der Zahlen auf dem Sudoku gibt es mehrere Funktionen um eine gute UX zu erreichen.

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

5.2.1. Funktionalitäten NumPad und andere Eingaben

Es gibt auf dem UI ein NumPad mit verschiedenen Tasten. Wenn das Sudoku noch nicht eingetragen ist, unterscheidet sich die Anordnung der Knöpfe. Am Anfang sind in der obersten Reihe auf dem NumPad die Tasten Start, Clear everything (CE) und Del zu sehen.

Beim Drücken des Startknopf verschwindet dieser und wird durch einen Note Knopf ersetzt. Zudem erscheint unter dem NumPad ein Help Knopf. Die Zahlen die zum Zeitpunkt des Drücken eingegeben wurden, bekommen eine andere Farbe und sind fest auf dem Board eingetragen. Diese Zahlen können also ab diesem Zeitpunkt nicht mehr überschrieben werden. Mit der Taste CE wird das komplette Sudokuboard zurückgesetzt. Mit eingeschlossen ist damit der Reset vom Starten des Sudokus und der Help Knopf verschwindet. Wenn das Sudoku noch nicht gestartet wurde, werden einfach die Zahlen auf dem Board entfernt. Mit der Del Taste werden in den ausgewählten Zellen der Eintrag gelöscht. Für alle Knöpfe wurden auch Tasten auf der Tastatur definiert. *TODO: Welche tasten welche funktionalitäten* Mit der Note Taste wir der aktuelle Stand der Kandidaten angezeigt. Durch das mehrmalige Drücken des Help Knopf werden die abgestuften Hilfestellung angezeigt.

TODO: Zwei Bilder nebeneinander von verschiedenen NumPads

Mit den Ziffern des NumPad können ganz normal Zahlen in das Sudoku eingetragen werden. Für die bessere Nutzung an einem Computer wurde zudem die Eingabe über das NumPad auf der Tastatur oder über die obere Zahlenreihe auf der Tastatur eingegeben werden.

5.2.2. Funktionalitäten Board

Auch für das Board gibt es mehrere Funktionalitäten. Es gibt mehrere Möglichkeiten eine oder mehrere Zellen einzutragen. Um die Zelle auf der man sich aktuell befindet ist eine bunte Umrandung. Diesen Curser kann man mit den Pfeiltasten bewegen. Wenn eine Zahl gedrückt wird, dann wird diese immer in der Zelle eingetragen auf der sich der Curser gerade befindet. Sobald die Zahl eingetragen wird springt der Cursor auf die nächste Zelle. Diese Funktionalität wurde auch mit der Tabulator Taste realisiert Es gibt aber auch die Möglichkeit durch Leertaste oder Mausklick mehrere Zellen auszuwählen. Die ausgewählten Zellen sind dann grau gefärbt. Wenn man die Markierung einer Zelle wieder rückgängig machen möchte, ist dies durch nochmaliges klicken möglich.

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

In der Abbildung xx sieht man das der Cursor ganz oben links auf der ersten Zelle (1, A) ist. Zudem sind mehrere Felder ausgewählt und daher grau hinterlegt.

TODO: Bild mit Markierungen

5.3. Abstufungen der Hilfestellungen

Die Anforderung an die verschiedenen Abstufungen der Hilfestellung wird hauptsächlich über das Frontend geregelt. Im Backend werden dafür nur die relevanten Informationen gesammelt. Im Folgenden wird die Abstufung und Umsetzung der Hilfestellungen gezeigt und Erläutert.

5.3.1. Erste Hilfestellung

Die erste Hilfestellung besteht daraus die anzuwendende Technik auf dem Frontend anzuzeigen und relevante Zellen zu markieren. In dem Codebeispiel 5.1 ist die Umsetzung dafür beschrieben.

```
1  socket.on('help0', function (technique_result) {
2    $("#technique-name").text(technique_result['name']);
3    $("#technique-explanation").text("");
4    colorCells(technique_result['primaryCells'], 'rgb(255,216,115)');
5    colorCells(technique_result['secondaryCells'], 'rgb(181,216,244)');
6  });
```

Listing 5.1: Erste Hilfestellung

Es wird der Klasse *technique-name* die Technik übergeben und über colorCells können Zellen auf dem Board in verschiedenen Farben markiert werden. Diese Markierungen und das Anzeigen des Technik Namen ist in Abbildung xx dargestellt. Die Zellen können in unterschiedlich eingefärbt werden, abhängig davon wie relevant sie für eine Technik sind. In den meisten Fällen wirken sich die *primaryCells* auf eine Technik aus und sind darauf ausgelegt zuerst aufzufallen. Die secondaryCells haben aber auch eine Bedeutung für eine Technik weisen dabei aber eher auf den Grund hin, weshalb eine Technik angewendet werden kann oder in welchen Units die Technik gefunden wurde.

TODO: Screenshot umsetzung

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

5.3.2. Zweite Hilfestellung

Mit der zweiten Hilfestellung werden relevante Kandidaten markiert. Im Falle der Kandidaten gibt es erneut zwei unterschiedliche Markierungen. Ein Teil dieser Hilfestellung ist das Anzeigen der Kandidaten, für den Fall dass dies nicht bereits der Fall ist.

```
1  socket.on('help1', function (technique_result) {
2    if (!candidatesVisible) {
3      candidatesVisible = !candidatesVisible;
4      $('#candidate').css('color', 'red');
5    }
6    colorCandidates(technique_result['crossOuts'], 'red');
7    colorCandidates(technique_result['highlights'], 'lime');
8  });
```

Listing 5.2: Zweite Hilfestellung

Kandidaten die für eine Lösungstechnik Relevanz haben können mittels der Funktion *colorCandidates* markiert werden. Auch hier wird mit zwei Farben unterschieden. Kandidaten die aufgrund der Technik heraus gestrichen werden können, werden auf dem Sudokuboard Rot markiert. Kandidaten auf denen eine Technik aufbaut werden grün angezeigt. Die Umsetzung wird in dem Codebeispiel 5.2 beschrieben und das Ergebnis in der Abbildung xx dargestellt.

TODO: Screenshot umsetzung

5.3.3. Dritte Hilfestellung

Die dritte Hilfestellung besteht aus einer Erklärung der Technik anhand des aktuellen Sudokus. Die Erläuterung wird dabei unter dem Technikname in der Klasse *technique-explanation* in einem <p> Element angezeigt. Diese Hilfestellung wurde analog zu dem Technikname in dem Codebeispiel 5.1 umgesetzt.

5.3.4. Vierte Hilfestellung

Nachdem das Programm dem Nutzer über mehrere Schritte geholfen hat die Technik zu finden und gegebenenfalls nachzuvollziehen, wird mit der letzten Hilfestellungen die Technik ausgeführt und die rot eingefärbten Kandidaten vom Board entfernt. Außerdem

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

werden alle Markierungen mit der Funktion *resetCellColor()*, wie in dem Codebeispiel 5.3 zurückgesetzt und der Techniktitel und die Beschreibung entfernt.

```
1  socket.on('help3', function () {
2    if (!candidatesVisible) {
3      candidatesVisible = !candidatesVisible;
4      $('#candidate').css('color', 'red');
5    }
6    $("#technique-name").text("");
7    $("#technique-explanation").text("");
8    resetCellColor();
9  });
```

Listing 5.3: Vierte Hilfestellung

5.4. Fertig gelöstes Sudoku

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

6. Backend

6.1. Programmarchitektur

TODO: UML

6.2. Serverseite

```
1   @socketio.on('help')
2   def help():
3   global help_step
4   global technique_result
5   errors = sudoku.get_errors()
6   if errors:
7   print(errors)
8   emit('showErrors', errors)
9   return
10  if help_step == 0:
11  sudoku.update_candidates()
12  technique_result = technique_manager.try_techniques(sudoku.board,
        sudoku.candidates)
13  if not technique_result:
14  print("No suitable technique found!")
15  return
16  emit('help0',
17  {'name': technique_result['name'], 'primaryCells': technique_result['
        primary_cells'],
18    'secondaryCells': technique_result['secondary_cells']})
19  elif help_step == 1:
20  candidates()
21  emit('help1', {'highlights': technique_result['highlights'], '
        crossOuts': technique_result['cross_outs']})
22  elif help_step == 2:
```

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

```
23    emit('help2', {'name': technique_result['name'], 'explanation':
          technique_result['explanation']})
24    elif help_step == 3:
25    if technique_result['name'] in ['Naked Single', 'Hidden Single', '
          Third Eye']:
26    data = {'number': technique_result['highlights'][0]['value'],
27      'checkedCells': [technique_result['highlights'][0]['cell']]}
28    emit('help3')
29    new_numbers(data)
30    help_step -= 1
31    else:
32    sudoku.remove_candidates(technique_result['cross_outs'])
33    candidates()
34    emit('help3')
35    help_step += 1
36    help_step %= 4
```

Listing 6.1: Serverseitige Implementierung für das Anwenden der Hilfestellungen

6.3. SudokuBoard

6.3.1. Datenstruktur des Sudokuboards

0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8
1,0	1,1	1,2	1,3	1,4	1,5	1,6	1,7	1,8
2,0	2,1	2,2	2,3	2,4	2,5	2,6	2,7	2,8
3,0	3,1	3,2	3,3	3,4	3,5	3,6	3,7	3,8
4,0	4,1	4,2	4,3	4,4	4,5	4,6	4,7	4,8
5,0	5,1	5,2	5,3	5,4	5,5	5,6	5,7	5,8
6,0	6,1	6,2	6,3	6,4	6,5	6,6	6,7	6,8
7,0	7,1	7,2	7,3	7,4	7,5	7,6	7,7	7,8
8,0	8,1	8,2	8,3	8,4	8,5	8,6	8,7	8,8

Abbildung 6.1.: Struktur des Sudokuboards und Zellenidentifikation

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

6.3.2. Funktionalität

6.4. Technique Manager

In der Datei *technique_manager.py* wird das Aufrufen der Techniken geregelt. Alle Lösungsstrategien sind als Klassenrepräsentation der umgesetzten Technik in einer Liste gespeichert. Über diese Liste wird drüber iteriert und das Board und die Kandidaten übergeben. Danach wird die abstrakte Methode *execute_technique()* für die Lösungstechnik ausgeführt. Der Returnwert dieser Methode ist True, wenn sich etwas an dem Board oder an den Kandidaten verändert. In diesem Fall wird die Methode *get_result()* aufgerufen, mit der die Änderungen durchgeführt werden können. Wenn eine Technik nicht erfolgreich ist dann wird mit der nächsten Lösungsstrategie aus der Liste weiter gemacht.

Die Reihenfolge der Lösungstechniken ist von der Website [Mar] übernommen. Zuerst werden einfache Lösungsansätze ausprobiert und wenn diese nicht erfolgreich sind, werden immer komplexere Techniken angewendet.

```
1  def try_techniques(board, candidates):
2  for tech in techniques:
3  technique = tech(board, candidates)
4  successful = technique.execute_technique()
5  if successful:
6  return technique.get_result()
7  return False
```

Listing 6.2: Funktion um eine anwendbare Lösungstechnik zu finden

6.5. Abstrakte Klasse SolvingTechniques

TODO: Möglichmacher für Liste der Klassen, daher können alle auf selbe Art und weise instanziiert werden und gleiche funktionen.

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

6.5.1. Abstrakte Methoden

configure_highlighting(self)

execute_technique(self)

update_explanation(self)

6.5.2. Hilfsfunktionen

TODO: es gibt in der abstrakten Methode berits implementierte Hilfsfunktionen, Nutzung bei mehreren Lösungstechniken notwenidg/hilfreich

6.6. Lösungsstrategien

In diesem Abschnitt wird ein Überblick über die implementierten Strategien gegeben. Die Lösungsstrategien sind auf der Website von Simon Martin [Mar] betitelt und beschrieben. Anhand diesen Beschreibungen wurden einige Lösungsstrategien implementiert. Zunächst wird ein Überblick über die umgesetzten Strategien gegeben, bevor ausblickend die Lösungsstrategien, die noch nicht implementiert wurden, aufgezählt werden. Es gibt noch weitere Strategien die auf der Website nicht beschrieben werden. Diese werden im Folgenden aber nicht weiter beachtet.

6.6.1. Umgesetzte Lösungsstrategien

Da in der Ausarbeitung nur exemplarisch auf einige Lösungsstrategien eingegangen wird, wird hier aufgezählt welche Strategien alle umgesetzt wurden.

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

Technikname	Technikname engl.
Versteckter Single	Hidden Single
Verstecktes Paar	Hidden Pair
Versteckter Dreier	Hidden Triple
Verteckter Vierer	Hidden Foursome
Reihe-Block-Check (RBC)	Line-Block-Ineraction
X-Wing	X-Wing
Drittes Auge	Third Eye
Schwertfisch	Swordfish
Verbotenes Rechteck Typ 1	Forbidden Rectangle Type 1
Verbotenes Rechteck Typ 3	Forbidden Rectangle Type 3
XY-Wing	XY-Wing

Technikname	Technikname engl.
Nackter Single	Naked Single
Nacktes Paar	Naked Pair
Nackter Dreier	Naked Triple
Nackter Vierer	Hidden Foursome
Block-Reihe-Check (BRC)	Block-Line-Interaction
Steinbutt	Turbot
Wolkenkratzer	Skyscraper
Drachen	Dragon
Verbotenes Rechteck Typ 2	Forbidden Rectangle Type 2
Verbotenes Rechteck Typ 4	Forbidden Rectangle Type 4

Tabelle 6.1.: Aufzählung alles implementierten Lösungsstrategien mit deutschem und englischem Bezeichner

6.6.2. Weitere Lösungsstrategien

In der Studienarbeit wurden nicht alle auf der Website [Mar] vorgestellten Lösungstechniken umgesetzt. Weitere Strategien die noch implementiert werden könnten sind die folgenden:

- Erweiterter BRC
- (noch nicht) Verbotenes Rechteck Typ 3
- (noch nicht) XY-Wing
- (noch nicht) XYZ-Wing
- (noch nicht) X-Kette
- (noch nicht) XY-Kette
- (noch nicht) Schwertfisch mit Flosse
- (noch nicht) W-Wing
- (noch nicht) Geklonte Paare
- (noch nicht) Leeres Rechteck
- (noch nicht) Doppelkette

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

- (noch nicht) Forcing Chain

Der Erweiterte BRC wurde nicht umgesetzt, weil es eine Kombination aus dem BRC und einem Versteckten Single ist.

6.7. Beispielhafte Umsetzung von Strategien

Die Lösungsstrategien können zwei unterschiedliche Wirkungen auf ein Sudokurätsel haben. Entweder es ist möglich eine Zahl einzufügen oder es können Kandidaten aus verschiedenen Gründen eliminiert werden. Diese beiden Wirkungen werden im Backend unterschiedlich behandelt. In den nächsten zwei Unterkapiteln wird die Funktionsweise und die Unterschiede beider Varianten anhand zweier Beispiele erklärt.

Für beide Umsetzungen wird die Abstrakte Klasse *SolvingTechniques* genutzt.

6.7.1. Zahl einfügen

Der Fall, dass aufgrund einer Lösungstechnik direkt eine Zahl in das Sudokugitter eingetragen werden kann, gibt es nur bei drei Techniken. Für alle anderen Techniken ist das eintragen einer Zahl maximal das Resultat von dem Herausstreichen einer Zahl.

- Versteckter Single
- Nackter Single
- Drittes Auge

6.7.2. Kandidat eliminieren

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

7. Testing/Validierung

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

8. Fazit

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

Literatur

[AV14] I. Althöfer und R. Voigt. *Spiele, Rätsel, Zahlen: Faszinierendes zu Lasker-Mühle, Sudoku-Varianten, Havannah, EinStein würfelt nicht, Yavalath, 3-Hirn-Schach, ...* Springer Berlin Heidelberg, 2014. ISBN: 9783642553011. URL: https://books.google.de/books?id=5vxXBAAAQBAJ.

[Ern20] P. Ernesti J. und Kaiser. *Python 3, Das umfassende Handbuch.* 5., aktualisierte Auflage. Bonn: Rheinwerk Computing, 2020. ISBN: 978-3-8362-7926-0.

[FJ06] Bertram Felgenhauer und Frazer Jarvis. „Mathematics of Sudoku I“. In: *Mathematical Spectrum* 39 (Jan. 2006).

[Her22] Michael Herman. *Django vs. Flask in 2022: Which framework to choose.* Website. Verfügbar unter https://testdriven.io/blog/django-vs-flask/; eingesehen am 25.05.2022. 2022.

[HM07] Agnes Herzberg und Ram Murty. „Sudoku squares and chromatic polynomials“. In: *Internationale Mathematische Nachrichten* 54 (Juni 2007).

[Ion] *Nginx vs. Apache: Die beliebtesten open-source-webserver im Vergleich.* Website. Verfügbar unter https://www.ionos.de/digitalguide/server/knowhow/nginx-vs-apache-ein-webserver-vergleich/; eingesehen am 25.05.2022. 2017.

[Kar19] Florian Karlstetter. *Was ist ein webserver?* Website. Verfügbar unter https://www.cloudcomputing-insider.de/was-ist-ein-webserver-a-884026/; eingesehen am 25.05.2022. 2019.

[Kno17] Simon Knott. *Backtracking.* Website. Verfügbar unter https://simonknott.de/articles/backtracking; eingesehen am 25.05.2022. 2017.

[Log14] D. Logofătu. *Grundlegende Algorithmen mit Java: Lern- und Arbeitsbuch für Informatiker und Mathematiker.* Springer Fachmedien Wiesbaden, 2014. ISBN: 9783834819727. URL: https://books.google.de/books?id=GBcyngEACAAJ.

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

[Lub22] Stefan Luber. *Was ist ein Framework?* Website. Verfügbar unter https://www.cloudcomputing-insider.de/was-ist-ein-framework-a-1104630/; eingesehen am 25.05.2022. 2022.

[Mar] Simon Martin. *Sudoku*. Website. Verfügbar unter https://www.thinkgym.de/rätselarten/sudoku/; eingesehen am 26.05.2022.

[Mus21] O. Musch. *Design Patterns mit Java: Eine Einführung*. Springer Fachmedien Wiesbaden, 2021. ISBN: 9783658354916. URL: https://books.google.de/books?id=nreUzgEACAAJ.

[Sar20] Dejan Sarka. *A Python Data Analyst's Toolkit, Learn Python and Python-based Libraries with Applications in Data Analysis and Statistics*. 6., aktualisierte Auflage. Berkeley, CA: Apress, 2020. ISBN: 978-1-4842-7172-8.

[Sta] *Stack overflow developer survey 2021*. Website. Verfügbar unter https://insights.stackoverflow.com/survey/2021; eingesehen am 25.05.2022.

[Ste20] R. Steyer. *JavaScript Grundlagen*. HERDT, 2020. ISBN: 978-3-86249-957-1.

[Sud22] Sudopedia. Website. Verfügbar unter http://www.sudopedia.org/; eingesehen am 25.05.2022. 2022.

[W3s] *jQuery Introduction*. Website. Verfügbar unter https://www.w3schools.com/jquery/jquery_intro.asp; eingesehen am 25.05.2022.

[W3t] *Usage statistics of web servers*. Website. Verfügbar unter https://w3techs.com/technologies/overview/web_server; eingesehen am 25.05.2022.

[Zam15] G. Zambon. *Sudoku Programming with C*. Apress, 2015. ISBN: 9781484209950. URL: https://books.google.de/books?id=xb4ICAAAQBAJ.

[Zei12] Frankfurter Allgemeine Zeitung. *Mathematik: Der heilige gral der sudokus*. Website. Verfügbar unter https://www.faz.net/aktuell/wissen/physik-mehr/mathematik-der-heilige-gral-der-sudokus-11682905.html; eingesehen am 27.05.2022. 2012.

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.

Anhang

PEA4-Fe – betriebliche Ausbildung technische Studiengänge der DHBW © Alle Rechte bei Robert Bosch GmbH, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht bei uns.