

Realisierung eines Vier-Gewinnt Roboter

Mithilfe von Lego Spike und Mirco Python

Studienarbeit T3_3100

Studiengang Elektrotechnik

Studienrichtung Automation

Duale Hochschule Baden-Württemberg Ravensburg, Campus Friedrichshafen

von

Patrik Peters / Simon Gschell

| | |
|------------------------|------------------------------|
| Abgabedatum: | 9. Januar 2025 |
| Bearbeitungszeitraum: | 10.10.2024 - 13.06.2025 |
| Matrikelnummer: | 187 /0815 |
| Kurs: | TEA22 |
| Betreuerin / Betreuer: | Prof. Dr. ing Thorsten Kever |

Erklärung

gemäß Ziffer 1.1.14 der Anlage 1 zu §§ 3, 4 und 5 der Studien- und Prüfungsordnung für die Bachelorstudiengänge im Studienbereich Technik der Dualen Hochschule Baden-Württemberg vom 29.09.2017 in der Fassung vom 24.07.2023.

Ich versichere hiermit, dass ich meine Studienarbeit T3_3100 mit dem Thema:

Realisierung eines Vier-Gewinnt Roboter

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.

Musterstadt, den 9. Januar 2025

Patrik Peters / Simon Gschell

Kurzfassung

In der vorliegenden Studienarbeit wird die Entwicklung eines Roboters für das Spiel „Vier gewinnt“ unter Verwendung des LEGO Spike Prime Systems behandelt. Ziel des Projekts ist es, einen Roboter zu entwerfen, der in der Lage ist, autonom die Rolle eines menschlichen Gegners im Spiel „Vier gewinnt“ zu übernehmen. Der Roboter soll nicht nur die Spielzüge des menschlichen Mitspielers erkennen und darauf reagieren, sondern auch selbstständig Spielsteine in das Spielfeld einwerfen und sicherstellen, dass das Spielfeld für den nächsten Zug bereit ist. Dies erfordert eine präzise Steuerung des Roboters, insbesondere beim Platzieren der Spielsteine, sowie die Fähigkeit, das Spielfeld zu überwachen, um die Position der bereits gesetzten Steine zu erkennen. Das Projekt umfasst verschiedene technische und organisatorische Aspekte. Dazu gehört die mechanische Konstruktion des Roboters, einschließlich des Mechanismus zum Einwerfen der Spielsteine und die Überwachung des Spielfelds, sowie die Entwicklung der Software, die die Spielzüge und das Verhalten des Roboters steuert. Ein weiterer wichtiger Punkt ist die Sensorik: Der Roboter muss in der Lage sein, die aktuelle Spielsituation durch Farbsensoren zu erfassen, um zu wissen, welche Felder im Spielfeld bereits besetzt sind und wo er seinen nächsten Spielstein platzieren kann. Zum Abschluss des Projekts wird ein kleines Turnier organisiert, bei dem die von verschiedenen Teams entwickelten Roboterlösungen gegeneinander antreten. Dies bietet die Möglichkeit, den Roboter unter realen Bedingungen zu testen und seine Fähigkeiten im direkten Vergleich mit den Lösungen der anderen Kommilitonen zu messen. Um zusätzliche Motivation zu schaffen, wird die beste Lösung am Ende prämiert, was den Wettbewerbsgedanken fördert und den Anreiz erhöht eine effektive Lösungen zu entwickeln.

Abstract

This student research project deals with the development of a robot for the game “Four Wins” using the LEGO Spike Prime system. The aim of the project is to design a robot that is capable of autonomously assuming the role of a human opponent in the game “Four Wins”. The robot should not only recognize the moves of the human opponent and react to them, but also independently place pieces on the playing field and ensure that the playing field is ready for the next move. This requires precise control of the robot, especially when placing the pieces, as well as the ability to monitor the playing field in order to recognize the position of the pieces already placed. The project involves various technical and organizational aspects. These include the mechanical design of the robot, including the mechanism for inserting the tiles and monitoring the playing field, as well as the development of the software that controls the moves and behavior of the robot. Another important point is the sensor technology: the robot must be able to detect the current game situation using color sensors in order to know which squares in the playing field are already occupied and where it can place its next piece. At the end of the project, a small tournament is organized in which the robot solutions developed by different teams compete against each other. This offers

Translated with DeepL.com (free version)

Inhaltsverzeichnis

| | | |
|----------|--|-----------|
| 1 | Einleitung | 1 |
| 2 | Grundlagen des Spiels Vier Gewinn | 2 |
| 2.1 | Spielregeln und Spielablauf | 2 |
| 2.2 | Historischer Hintergrund | 3 |
| 2.3 | Mathematische Eigenschaften des Spiels | 5 |
| 3 | Spieltheoretische Grundlagen | 6 |
| 3.1 | Definition und Konzepte der Spieltheorie | 6 |
| 3.2 | Klassifikation von Spielen | 7 |
| 3.3 | Relevante spieltheoretische Konzepte für Vier Gewinn | 10 |
| 4 | Analyse von Vier Gewinn aus spieltheoretischer Sicht | 12 |
| 4.1 | Darstellung des Spiels in Normalform | 12 |
| 4.2 | Komplexität des Spielbaums | 13 |
| 4.3 | Darstellung des Spiels in Extensivform | 14 |
| 4.4 | Strategien und Gleichgewichte | 15 |
| 5 | Lösungsansätze für Vier Gewinn | 16 |
| 5.1 | Optimale Strategien | 16 |
| 5.2 | Heuristische Ansätze | 18 |
| 5.3 | Entwicklung von Algorithmen | 20 |
| 5.3.1 | MinMax-Algorithmus | 20 |
| 5.3.1 | AlphaBeta-Algorithmus | 23 |
| 6 | Anwendung der Spieltheorie auf Vier Gewinn | 27 |
| 6.1 | Analyse von Spielsituationen | 27 |
| 6.2 | Berechnung von Gewinnwahrscheinlichkeiten | 28 |
| 6.3 | Entwicklung von Spielstrategien | 28 |
| 6.4 | Systementwurf | 32 |
| 6.4.1 | Komponenten | 32 |
| 6.4.2 | Zusammenspiel von Soft- und Hardware | 34 |

| | |
|---|-----------|
| 6.5 Mechanische Konstruktion des Farbsensor-Systems | 34 |
| 7 Zusammenfassung | 36 |
| Literaturverzeichnis | 37 |
| Abbildungsverzeichnis | 39 |
| Tabellenverzeichnis | 40 |

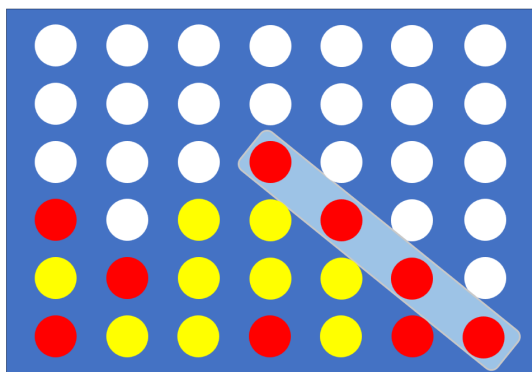
1 Einleitung

2 Grundlagen des Spiels Vier Gewinnt

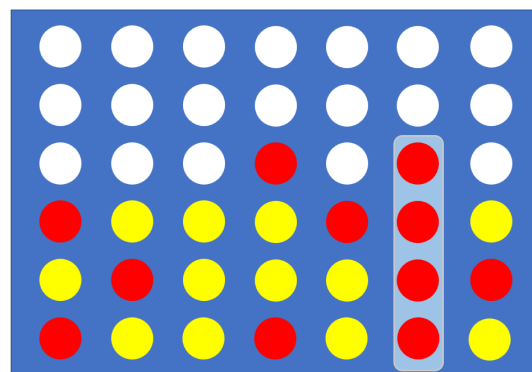
Im Kapitel Grundlagen wird auf die Regeln des Spiels, sowie auf den geschichtlichen Hintergrund als auch die mathematischen Eigenschaften des Spiels 4Gewinnt genauer eingegangen.

2.1 Spielregeln und Spielablauf

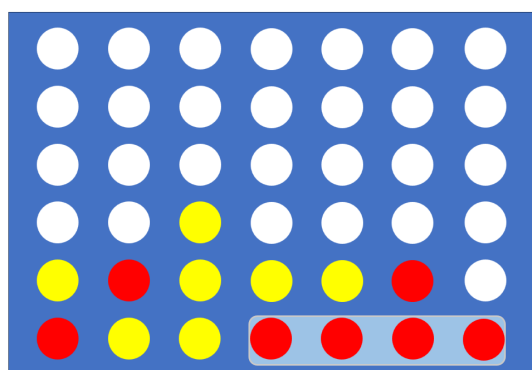
Das originale Spiel 4Gewinnt besteht aus einer Rasterwand mit sechs Zeilen und sieben Spalten, also 42 Löchern. Außerdem besteht es aus 21 roten Spielchips und 21 gelben Spielchips. 4Gewinnt lässt sich nur zu zweit spielen. Ziel des Spieles ist es, vier Spielchips einer Farbe in eine Reihe (waagrecht, senkrecht oder diagonal) zu bringen. Der jüngste Spieler beginnt das Spiel. Der Spieler, der an der Reihe ist, wirft einen Spielstein seiner Farbe durch die Öffnung der Rasterwand, wo durch der gespielte Stein auf den untersten freien Platz in der Spalte fällt. Danach ist der andere Spieler an der Reihe. Die Spieler werfen so lange ihre Spielchips in das Raster, bis einer das Ziel erreicht hat oder alle 42 Felder belegt sind. Für den Fall, dass alle 42 Felder des Rasters belegt sind, endet das Spiel unentschieden. Um ein neues Spiel zu starten, muss ein Spieler die Steine aus dem Raster fallen lassen. Diese werden dann wieder unter den Spielern aufgeteilt und eine neue Runde 4Gewinnt kann beginnen[Has20].



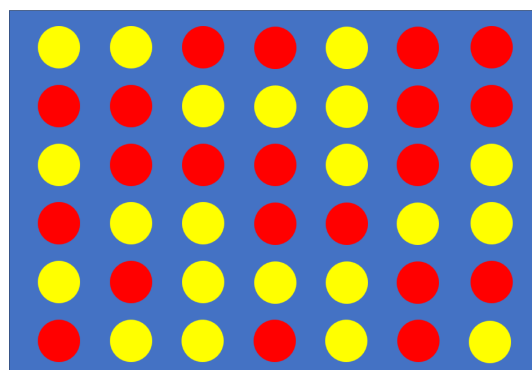
(a) Vier rote Steine diagonal im Raster



(b) Vier rote Steine senkrecht im Raster



(c) Vier rote Steine waagrecht im Raster



(d) Es kam keine vierer Reihe zustande

Abbildung 2.1: Grafische Darstellung der möglichen Spielausgänge

2.2 Historischer Hintergrund

Vier Gewinn wurde 1973 von Howard Wexler und Ned Strongin entwickelt. Noch im selben Jahr lizenzierten die beiden Erfinder das Spiel an die Firma Milton Bradley, im deutschsprachigen Raum besser bekannt als „MB Spiele“ [Gam25][Wik25]. Im darauffolgenden Jahr 1974 wurde das Spiel Vier Gewinn von Milton Bradley veröffentlicht und in den Handel gebracht. Das Spiel entwickelte sich seitdem zu einem der bekanntesten und beliebtesten Strategiespiele [Wik25]. Tatsächlich entwickelt es sich sogar zu einem echten Klassiker. Das Spiel ist beliebt bei Jung und Alt, da es recht leicht zu erlernen ist. Vier Gewinn fördert die Konzentration und das logische Denken [50P25].

Im Jahr 1967 erschien in den USA eine dreidimensionale Variante vom Spiel Vier Gewinnt, unter dem Namen „Score Four”. In Deutschland brauchte die Firma Ravensburger 1974 eine dreidimensionale Variante mit dem Namen „Sogo” auf den Markt. (siehe Abbildung:2.2) 1988 wurde das Spiel von Victor Allis und James D. Allen nahezu gleichzeitig und unabhängig voneinander vollständig gelöst. Die Beiden haben mathematisch bewiesen, dass der erste Spieler bei fehlerfreiem Spiel immer gewinnen kann, wenn er den ersten Stein in der mittleren Spalte platziert.[Wik25].



Abbildung 2.2: dreidimensionale Variante „Sogo” von Ravensburger

2.3 Mathematische Eigenschaften des Spiels

Das Spiel 4Gewinnt weist mehrere mathematische Eigenschaften auf. Diese sind für eine Analyse des Spiels von Bedeutung. Das Strategiespiel ist ein Zwei-Personen-Nullsummenspiel. Das bedeutet, es spielen zwei Teilnehmer gegeneinander. Beide Spieler haben während des Spiels immer die volle Information über den Spielstand. Wenn einer von beiden gewinnt, bedeutet das gleichzeitig, dass der andere verloren hat. Trotz der simplen Spielregeln ist vier Gewinnt dennoch recht komplex. Grund dafür die vielen möglichen Spielstellungen, es gibt nämlich ca. 4,5 Billionen mögliche Spielstellungen. Diese enorm hohe Anzahl an möglichen Spielstellungen kommt durch das Spielfeld mit sieben Spalten und sechs Reihen zustande. Die Komplexität des Baumdiagramms liegt bei etwa 10^{21} , was eine vollständige Analyse des Spielbaums sehr rechenintensiv macht [Rui+09].

Das Lösen des Spiels Vier Gewinnt basiert auf verschiedenen mathematischen Techniken. Zum einen gibt es die Anwendung des Zermelo-Bestimmtheitssatzes. Dieser besagt, dass entweder Spieler 1, Spieler 2 oder kein Spieler die Partie gewinnt. Im Kapitel 6.1 wird näher auf den Bestimmtheitssatz eingegangen wird.

Eine weite Technik zur Analyse ist die Anwendung der Rückwärtsinduktion. Die Rückwärtsinduktion beginnt an der aktuellen Position und arbeitet sich zum Anfang zurück. Hierbei werden für jeden Spielzug die optimalen Entscheidungen der Spieler analysiert, unter der Berücksichtigung aller möglichen Züge. Am Ende dieses Prozesses steht das beste Ergebnis für den beginnenden Spieler [Mül11].

Die Rückwärtsinduktion wird durch die Payoff-Matrix-Analyse ergänzt. Das grundlegende Konzept der Payoff-Matrix-Analyse bietet eine Möglichkeit zur Darstellung strategischer Interaktionen und Entscheidungen dar [Fas24].

| | | Spieler 2 | | |
|-----------|----------|-----------|---------|----------|
| | | Zentrum | Rand | Defensiv |
| Spieler 1 | Zentrum | (0, 0) | (1, -1) | (0, 0) |
| | Rand | (-1, 1) | (0, 0) | (-1, 1) |
| | Defensiv | (0, 0) | (1, -1) | (0, 0) |

Abbildung 2.3: Payoff-Matrix für Vier gewinnt

Legende der Payoff-Matrix

Rot = Spieler 1, Blau = Spieler 2,
1=Vorteil, 0=Neutral, -1=Verlust

3 Spieltheoretische Grundlagen

In diesem Kapitel werden die grundlegenden Konzepte der Spieltheorie eingeführt, die als mathematische Grundlage für strategische Entscheidungsfindung dienen. Zunächst erfolgt eine Definition der Spieltheorie sowie eine Klassifikation der verschiedenen Spielarten, die für diese Studienarbeit relevant sind. Anschließend werden zentrale spieltheoretische Konzepte erläutert, die sowohl allgemein anwendbar sind als auch speziell auf das Spiel 4 Gewinnt zugeschnitten werden.

3.1 Definition und Konzepte der Spieltheorie

Die Spieltheorie ist ein Teilgebiet der Mathematik, das strategische Entscheidungen in Konflikt- oder Kooperationssituationen analysiert. Ein Spiel besteht dabei aus mehreren Elementen: den Spielern, den möglichen Zügen, den Regeln und dem Ergebnis. Ziel der Spieltheorie ist es, optimale Strategien zu entwickeln und die möglichen Ergebnisse eines Spiels vorherzusagen, wenn die Spieler rational handeln.

Grundelemente eines Spiels sind folgende Aspekte:

1. **Spieler:** Die Teilnehmer eines Spiels, die Entscheidungen treffen. Es können zwei oder mehr Mitspieler beteiligt sein.
2. **Strategien:** Die Handlungsoptionen, die den Spielern zur Verfügung stehen, um ihre Ziele (Sieg gegen den Mitspieler) zu erreichen.
3. **Regeln:** Die Struktur des Spiels legt fest, wie die Züge ablaufen, in welcher Reihenfolge sie gemacht werden und unter welchen Bedingungen das Spiel endet.
4. **Auszahlungen:** Die Ergebnisse des Spiels werden für jeden Spieler anhand einer bestimmten Kombination von Strategien festgelegt. Diese Ergebnisse können in

Form von Gewinnen, Verlusten oder Nutzen dargestellt werden.

5. **Informationen:** Spiele können sich darin unterscheiden, ob die Spieler alle Informationen haben (das heißt, sie sehen alle Züge) oder ob es Unsicherheiten gibt (wie zum Beispiel verdeckte Karten oder Würfelergebnisse).[Hol19].

3.2 Klassifikation von Spielen

Um eine Strategie von 4Gewinnt umsetzen zu können, muss das Spiel an sich erstmal analysiert werden. In der Spieltheorie werden Spiele nach bestimmten Kriterien eingeteilt, um ihre Struktur, Dynamik und Lösbarkeit besser zu erfassen. Die wichtigsten Kategorien für diese Einordnung sind im Folgenden aufgeführt.

1. Anzahl der Spieler

- **Zwei-Personen-Spiele:** Spiele mit genau zwei Spielern. Solche Spiele werden oft spieltheoretisch analysiert.
- **Mehrspieler-Spiele:** Spiele mit drei oder mehr Teilnehmern. Die Strategien sind komplexer, da Bündnisse und wechselnde Zusammenschlüsse eine Rolle spielen können.

2. Spielzüge: gleichzeitige vs. aufeinanderfolgende Entscheidungen

- **Simultane Spiele:** Beide Spieler treffen ihre Entscheidungen gleichzeitig, ohne zu wissen, für welche Option der andere sich entschieden hat. Solche Spiele werden oft in einer Matrixform betrachtet, um sie besser zu analysieren.
- **Sequentielle Spiele:** Die Spieler machen ihre Züge nacheinander und haben die Möglichkeit, auf die Aktionen des Gegners zu reagieren. Um die Reihenfolge der Züge und die verschiedenen Entscheidungsmöglichkeiten anschaulich darzustellen, werden solche Spiele oft in Form eines Baumdiagramms, also einer Extensivform, visualisiert.

3. Informationen der Spieler

- **Spiele mit vollständiger Information:** Alle Spieler sind über die bisherigen

Züge und den aktuellen Stand des Spiels informiert. In solchen Situationen lassen sich optimale Strategien durch Rückwärtsinduktion erarbeiten.

- **Spiele mit unvollständiger Information:** Mindestens ein Spieler hat keine vollständigen Informationen über den Spielzustand oder die Strategie des Gegners. Dies führt zu Unsicherheiten bei der Entscheidungsfindung.

4. Kooperative vs. nicht-kooperative Spiele

- **Kooperative Spiele:** Spieler haben die Möglichkeit, Zusammenschlüsse zu vereinbaren und gemeinsam zu handeln, um sich einen Vorteil zu verschaffen. Dabei spielen Verhandlungen und Absprachen eine entscheidende Rolle für die Ergebnisse.
- **Nicht-kooperative Spiele:** Jeder Spieler agiert unabhängig und verfolgt ausschließlich seine eigenen Ziele.

5. Gewinnstruktur: Nullsummenspiele vs. Nicht-Nullsummenspiele

- **Nullsummenspiele:** Der Gewinn eines Spielers ist genau der Verlust des anderen. Diese Art von Spielen ist konfliktbeladen, weil der Vorteil für einen Spieler immer mit einem Nachteil für den anderen einhergeht.
- **Nicht-Nullsummenspiele:** Die Gewinne und Verluste der Spieler müssen nicht unbedingt übereinstimmen. In solchen Spielen kann es vorkommen, dass die Teilnehmer zusammenarbeiten oder ihre Gewinne teilen.

6. Deterministische vs. stochastische Spiele

- **Deterministische Spiele:** Es gibt keine Zufälle im Spiel, alles hängt ganz von den Entscheidungen der Spieler ab.
- **Stochastische Spiele:** Zufallselemente wie Würfel oder Karten bringen unvorhersehbare Wendungen ins Spiel, was für zusätzliche Spannungen und Unsicherheiten sorgt.

7. Endliche vs. unendliche Spiele

- **Endliche Spiele:** Die Anzahl der möglichen Spielzustände und Züge ist be-

grenzt. Das bedeutet, dass man solche Spiele komplett durchleuchten und die besten Strategien finden kann.

- **Unendliche Spiele:** Spiele, bei denen es potenziell unendlich viele Züge oder Zustände gibt und die sich nicht vollständig analysieren lassen.

Kriterien angewandt auf 4Gewinnt

Anhand diesen genannten Kriterien kann 4Gewinnt klassifiziert werden. 4Gewinnt ist nach den aufgelisteten fünf Kriterien ein kombinatorisches Spiel. Sie bieten einen idealen Rahmen, um Strategien, Entscheidungsfindung und Spieltheorie zu untersuchen. Auch weitere Spiele wie Schach, Dame, Mühle und Tic-Tac-Toe sind ebenfalls kombinatorische Spiele.

Diese folgende fünf Eigenschaften machen ein kombinatorisches Spiel aus:

1. **Zwei-Personen-Spiel:** 4Gewinnt ist ein klassisches Spiel für zwei Personen, bei dem die Spieler abwechselnd ihre Steine in ein Gitter fallen lassen.
2. **Nullsummenspiel:** In "Vier Gewinnt" verfolgen die beiden Spieler gegensätzliche Ziele. Wenn einer von ihnen gewinnt, indem er vier Steine in einer Reihe hat, bedeutet das gleichzeitig, dass der andere verloren hat.
3. **endliches Spiel:** Das Spielfeld hat insgesamt 42 Felder, die sich auf 6 Reihen und 7 Spalten verteilen. Nach einer bestimmten Anzahl an Zügen wird das Feld immer vollständig gefüllt sein. Das bedeutet, dass jedes Spiel immer zu einem Ende kommt – entweder gewinnt ein Spieler oder es endet unentschieden, wenn keine Züge mehr möglich sind.
4. **vollständige Information:** Beide Spieler können jederzeit den aktuellen Spielstand genau im Blick behalten, da alle Spielsteine offen liegen. Es gibt keine geheimen Informationen, sodass sie ihre Strategien auf Grundlage aller bisherigen Züge entwickeln können.
5. **Determinismus:** Es gibt keine Zufälle, der Ausgang des Spiels hängt ganz von den Entscheidungen der Spieler ab. Jede Spielsituation ist klar und vorhersehbar,

solange man die Züge der Spieler kennt.

Durch diese genannten fünf Eigenschaften werden kombinatorische Spiele auch **endliches ZweiPersonen-Nullsummenspiel mit perfekter Information** genannt [Hol19].

3.3 Relevante spieltheoretische Konzepte für Vier Gewinn

Mit Kapitel 3.2 wurde dargestellt, dass das 4Gewinnt ein strategisches Zwei-Personen-Spiel mit vollständiger Information ist und somit mithilfe der Spieltheorie analysieren lässt. Einige zentrale Konzepte der Spieltheorie sind besonders relevant, um die Dynamik und Strategien von 4Gewinnt zu verstehen:

Dominante Strategien:

- Eine Strategie ist dominant, wenn sie unabhängig von den Zügen des Gegners optimal ist.
- In 4Gewinnt existieren situationsabhängige dominante Strategien.
- Die Kontrolle der Mittelspalte gilt als teilweise dominante Strategie, da sie zentrale taktische Vorteile bietet.

Gleichgewichtskonzepte:

- **Nash-Gleichgewicht:** Ein Zustand, in dem kein Spieler durch einseitiges Abweichen ihre Position verbessern kann.
- **Teilspielperfekte Gleichgewichte:** Strategien, die in allen Teilspielen optimal sind.
- **Minimax-Prinzip:** Eine Methode, bei der der Spieler versucht, den maximal möglichen Verlust zu minimieren.

Strategische Tiefe:

- mehrere Züge im Voraus denken ist entscheidend für den Erfolg.
- Erkennung und gezielte Nutzung von Zwangszügen und Zugfolgen.
- Entwicklung von Bedrohungspotenzialen und deren effektive Abwehr.

Positionsbewertung:

- Bewertung von Spielstellungen nach strategischen Kriterien zur Einschätzung der Spielstärke.
- Erkennung von entscheidenden Momenten, die das Ergebnis des Spiels beeinflussen können.
- Untersuchung von Gewinnmöglichkeiten und Risiken, um potenzielle Spielstrategien zu entwerfen.

Zugzwang-Situationen:

- Zugzwang beschreibt Situationen, in denen jeder mögliche Zug die eigene Position verschlechtert.
- Solche Situationen haben eine hohe strategische Bedeutung für die Spielführung.
- Es ist wichtig, gezielte Methoden anzuwenden, um den Gegner in eine schwierige Situation zu bringen und ihn unter Druck zu setzen.

Basierend auf den vorgestellten Konzepten kann nun eine Strategie erstellt werden, der dabei hilft, einen Algorithmus zu konzipieren. Dieser Algorithmus würde verschiedene Elemente der Spieltheorie und strategischen Überlegungen einbeziehen, um die Entscheidungsfindung zu verbessern.

4 Analyse von Vier Gewinnt aus spieltheoretischer Sicht

Die spieltheoretische Analyse des Spiels Vier Gewinnt erfordert eine systematische Untersuchung verschiedener strategischer Komponenten und mathematischer Aspekte. Im Folgenden Kapitel „Analyse von Vier Gewinnt aus spieltheoretischer Sicht“ werden die wesentlichen Elemente dieser Analyse detailliert dargestellt.

4.1 Darstellung des Spiels in Normalform

In der Normalform-Darstellung oder auch Normalform genannt, wird das Spiel als Zwei-Personen-Nullsummenspiel charakterisiert. Das bedeutet, wenn ein Spieler gewinnt, verliert der andere automatisch. Das Spiel wird in der Normalform rein statisch beschrieben und besteht allgemein aus folgenden Elementen[Sie]:

- Einer Menge Spieler ($I = 1, \dots, i, \dots, n$)
Der Wert I gibt die Anzahl der Teilnehmer an.
- Für jeden Spieler i eine Menge von Strategien (S_i)
(S_i) entspricht der Strategiemenge des Spielers i , aus dieser er seine Züge wählen kann.
- Für jeden Spieler i eine Auszahlungsfunktion/ Nutzenfunktion (u_i)
Hierbei wird jeder möglichen Strategiekombination aller Spieler einen reellen Zahlenwert zugeordnet.

$$u_i = \sum S_i \rightarrow \mathbb{R}^n \quad (4.1)$$

Ist die Normalform endlich und überschaubar, kann sie auch in einer Matrix dargestellt werden. Die Matrix muss dabei die verschiedenen Spielsituationen und deren Ausgänge abbilden, wobei jeder Spieler in seiner Zugfolge bis zu sieben verschiedene Strategien pro Zug zur Verfügung hat. Durch die sehr große Anzahl an Strategien ist es nicht mehr überschaubar und daher eher weniger geeignet.

4.2 Komplexität des Spielbaums

Die Komplexität des Spielbaums ist beträchtlich und wird oft als „mittlere Komplexität“ beschrieben. Die liegt daran, dass es eine große Anzahl an möglichen Spielzuständen gibt, aber das Spiel sich dennoch lösbar ist.

Die Anzahl der möglichen Konstellationen bei Vier Gewinnt beträgt laut einer Studie von John Tromp 4.531.985.219.092 (circa $4,5 \cdot 10^{12}$) [Thi12]. Aufgrund dieser enormen Zahl wird deutlich, warum eine vollständige Durchsuchung des Spielbaums für ein Vier-Gewinnt-Spiel eine Herausforderung darstellt.

Pro Zug gib es für den Spieler sieben Möglichkeiten. Bereits in der nach ersten Runde hat der Spielbaum $7 \cdot 7 = 49$ Äste, also 49 mögliche Spielzustände. Das zeigt, dass der Spielbaum trotz der scheinbaren Einfachheit des Spiels Vier Gewinnt, recht komplex ist [Thi12][Rui+09]. Aus diesem Grund wurden Ansätze entwickelt, die den Spielbaum mithilfe von verteilten Rechensystemen effizient lösen. Ein Beispiel hierfür ist „Shard Solver“ [YGJ22].

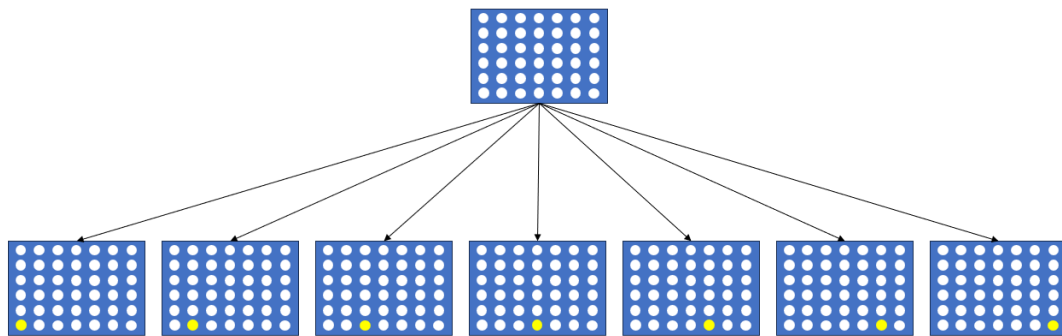


Abbildung 4.1: Spielbaum nach dem 1. Zug von Spieler 1

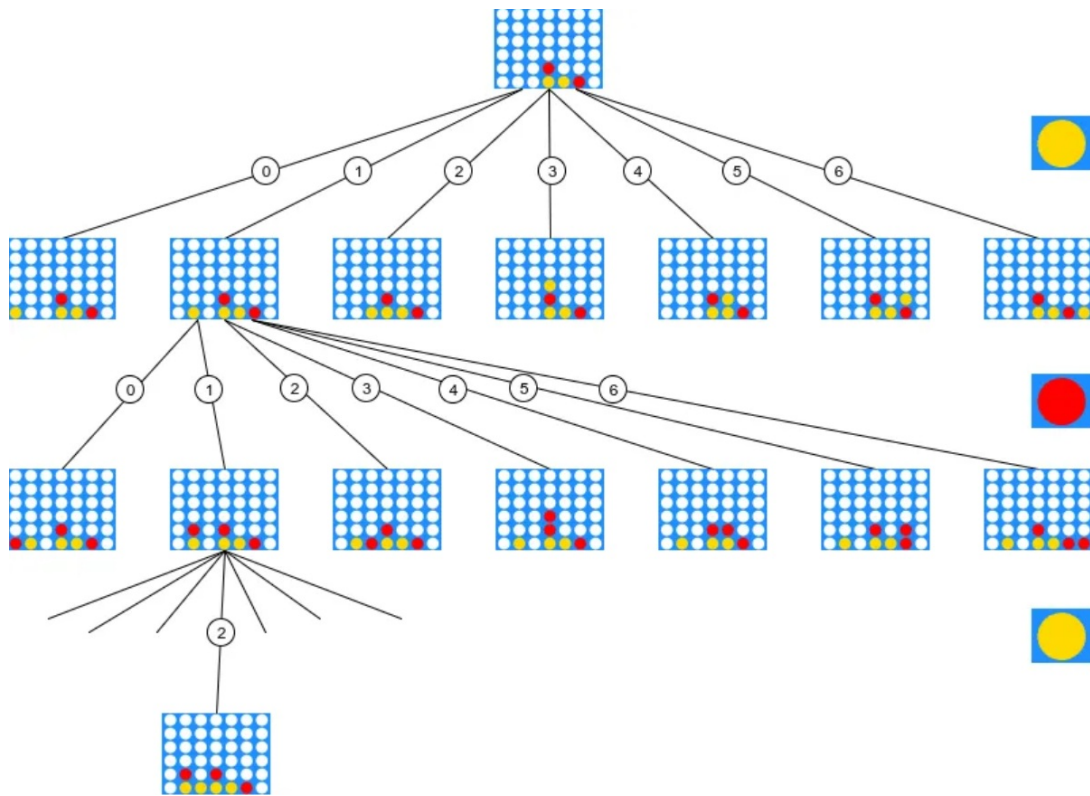


Abbildung 4.2: Schematische Darstellung des Spielbaum nach dem 4. bis zum 7. Zug

4.3 Darstellung des Spiels in Extensivform

Die Extensivform Darstellung ist eine aufschlussreiche Methode. Sie ermöglicht eine detaillierte Abbildung des Spielverlaufs und der Entscheidungsmöglichkeiten aller Spieler. In dieser Form wird ein Spielbaum verwendet, der aus Knoten und Kanten besteht [Ein14]. Der Baum beginnt mit einem Wurzelknoten, hier ist es ein leeres Spielfeld mit 42 freien Feldern. Die Verbindungen zwischen den Knoten, auch Kanten genannt, stellen die möglichen Züge für den Spieler dar. Die Knoten, an denen Kanten zusammenlaufen, repräsentieren eine Entscheidung eines Spielers. Die Knoten am Ende zeigen mögliche Ausgänge für das Spiel an. Bei Vier Gewinnt stellt jede Ebene des Spielbaums einen Spielzug dar. Pro Zug gibt es immer sieben Möglichkeiten, den Spielstein zu platzieren. Das bedeutet, von jedem Knoten gehen sieben Kanten weg. Allerdings ist der Baum durch die Spielfeldgröße auf 42 mögliche Züge begrenzt.

Um die Extensivform praktisch nutzen zu können, werden Vereinfachungen und Optimierungen durchgeführt. Eine Möglichkeit ist es, nutzlose Züge wegzulassen. Dadurch soll der Spielbaum vereinfacht werden und sich besser für Analysen eignen [Rui+09].

4.4 Strategien und Gleichgewichte

Die Analyse der Strategien zeigt, dass es im Vier Gewinnt dominante Strategien gibt. Nach dem Zermelo'schen Bestimmtheitssatz lässt sich das Spiel in eine von drei Kategorien einordnen, wobei sich herausgestellt hat, dass der erste Spieler eine dominante Strategie besitzt⁵. Die Nash-Gleichgewichte manifestieren sich in den optimalen Zugfolgen, wobei die Kontrolle der zentralen Spalten eine entscheidende Rolle spielt².

5 Lösungsansätze für Vier Gewinnt

5.1 Optimale Strategien

Die Analyse von 4Gewinnt aus der Perspektive der Spieltheorie zeigt eine Reihe von strategischen Prinzipien und optimalen Spielweisen um gezielte Strategien zu entwickeln. Diese Erkenntnisse beruhen sowohl auf grundlegenden Prinzipien, welche in Kapitel 3 aufgearbeitet wurden, als auch auf fortgeschrittenen Taktiken, die im Verlauf des Spiels entscheidend sein können.

Ein wichtiges Grundprinzip ist die Bedeutung des ersten Zuges. Es wurde festgestellt, dass der erste Spieler, wenn er perfekt spielt, immer gewinnen kann – vorausgesetzt, er wählt den besten Eröffnungszug. Der optimale Zug ist dabei, den ersten Stein in die Mitte zu setzen. Diese Eröffnung ermöglicht die beste Kontrolle über das Spielbrett und eröffnet viele Möglichkeiten für offensive und defensive Züge. Zieht der erste Spieler jedoch in eine der angrenzenden Spalten, führen beide Spieler bei optimalem Spiel zu einem Unentschieden. Alle anderen Eröffnungszüge gelten als weniger effektiv und führen, wenn der Gegner ebenfalls perfekt spielt, unweigerlich zur Niederlage des ersten Spielers.

Ein weiteres zentrales Prinzip ist die Kontrolle des Zentrums, die eine entscheidende Rolle spielt. Die Kontrolle über die mittleren drei Spalten (insbesondere c, d und e) ist strategisch besonders wichtig, da sie die besten Chancen für horizontale, vertikale und diagonale Gewinnreihen bietet. Auch die Spalten b und f – die zweite und vorletzte Spalte – sind wichtig, denn ohne sie kann der Spieler keine vollständigen diagonalen oder horizontalen Viererreihen bilden. Diese Kontrolle hilft nicht nur beim eigenen Spielaufbau, sondern schränkt auch die Möglichkeiten des Gegners ein.

Im Laufe des Spiels kommen zudem fortgeschrittene taktische Elemente ins Spiel. Eine wichtige Taktik ist die Schaffung von Zugzwang-Situationen, in denen der Gegner gezwungen wird, einen Zug zu machen, der seine eigene Position schwächt. Diese Situationen sind besonders am Ende des Spiels von Bedeutung, wenn nur noch wenige Felder

zur Verfügung stehen und der Druck auf beide Spieler steigt. Eine weitere wichtige Taktik sind Fallenkombinationen, bei denen der Gegner durch clevere Kombinationen von Bedrohungen in eine Falle gelockt wird. Besonders wirkungsvoll sind Doppelfallen, bei denen zwei gleichzeitige Drohungen entstehen, die der Gegner nicht gleichzeitig abwehren kann. Auch Auffüllfallen sind relevant: Hier zwingt ein Spieler den Gegner durch einen Mangel an freien Feldern dazu, einen entscheidenden Stein in eine vorbereitete Blockade zu setzen.

Die spieltheoretische Analyse von „Vier Gewinnt“ wurde durch mathematische und computerbasierte Methoden weiter vertieft. Unabhängig voneinander fanden zwei Forscher eine vollständige Lösung für das Spiel. Victor Allis entwickelte 1988 einen speziellen Regelsatz zur systematischen Analyse, während James D. Allen 1990 Computerprogramme einsetzte, um das Spiel vollständig zu berechnen. Beide kamen unabhängig zum gleichen Schluss: Der erste Spieler kann bei optimalem Spiel und einer Eröffnung in der Mitte immer gewinnen. Diese Erkenntnis hat nicht nur wissenschaftliche Bedeutung, sondern bildet auch die Basis für die Entwicklung von Algorithmen, die in Computerprogrammen oder Robotersystemen verwendet werden, um das Spiel optimal zu spielen. [Wik25]

Mögliche Startpositionen und Spielausgänge

1. Startposition 4 (mittlere Spalte) Wenn der erste Spieler in die mittlere Spalte setzt, hat er einen nachgewiesenen Vorteil und kann bei perfektem Spiel gewinnen. Diese Eröffnung maximiert die Kontrolle über das Spielfeld.

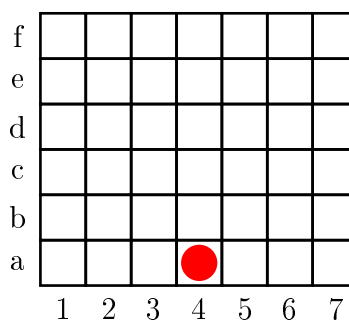


Abbildung 5.1: ROT gewinnt bei optimalen Spiel

2. Startpositionen 3 oder 5 (benachbarte Spalten zur Mitte) Ein Zug in die Spalte 3 oder 5 führt bei optimalem Spiel zu einem Remis, da der zweite Spieler durch eine Kombination aus Zentrums- und Blockstrategien den Sieg verhindern kann.

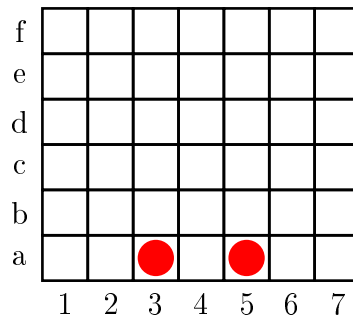


Abbildung 5.2: Remis bei optimalen Spiel

3. Startpositionen 1, 2, 6 oder 7 (äußere Spalten) Züge in die äußeren Spalten gelten als suboptimal. Der erste Spieler verliert bei perfektem Gegenspiel des zweiten Spielers, da diese Positionen weniger Kontrolle über das Zentrum und die Gewinnlinien bieten.

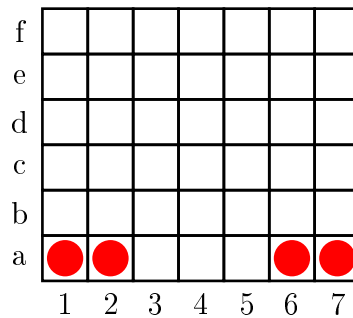


Abbildung 5.3: ROT verliert bei optimalen Spiel

Diese Analyse zeigt, wie wichtig die Wahl der Startposition für den weiteren Spielverlauf ist. Der erste Zug in die Mitte eröffnet dem Spieler die besten Chancen, während Züge in die äußeren Spalten zu deutlichen Nachteilen führen können.

5.2 Heuristische Ansätze

Die Bewertung von Positionen auf dem Spielfeld ist ein zentraler heuristischer Ansatz bei der Strategieentwicklung für 4Gewinnt. Dieser Ansatz zielt darauf ab, den strategischen Wert jeder Position zu analysieren und darauf aufbauend optimale Züge zu planen.

Die dargestellte Tabelle veranschaulicht die strategische Bewertung jedes Spielfelds. Jedes Feld erhält einen numerischen Wert, der angibt, wie viele mögliche Viererreihen dieses Feld beeinflussen kann. Ein Zug auf ein Feld mit einem höheren Wert ist in der Regel

strategisch besser, da er potenziell mehr Siegoptionen eröffnet. Solche Bewertungsansätze werden auch in computergesteuerten Spielen angewendet, um optimale Züge zu berechnen.

| | | | | | | | |
|---|---|---|----|----|----|---|---|
| f | 3 | 4 | 5 | 7 | 5 | 4 | 3 |
| e | 4 | 6 | 8 | 10 | 8 | 6 | 4 |
| d | 5 | 8 | 11 | 13 | 11 | 8 | 5 |
| c | 5 | 8 | 11 | 13 | 11 | 8 | 5 |
| b | 4 | 6 | 8 | 10 | 8 | 6 | 4 |
| a | 3 | 4 | 5 | 7 | 5 | 4 | 3 |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

Abbildung 5.4: Bewertungstabelle: mögliche Anzahl an Vierer-Reihen

Erklärung der Werte:

Zentrum des Spielfelds: Die Felder im Zentrum (insbesondere Spalte 4 und die Zeilen neben dran 3 und 5) haben die höchsten Werte, da sie Teil mehrerer potenzieller Viererreihen sein können – sowohl horizontal, vertikal als auch diagonal. Das erklärt, warum das Feld in Spalte 4, Zeilen c und d, den maximalen Wert von 13 besitzt.

Ränder des Spielfelds: Die Felder am Rand (also die ersten und siebten Spalten sowie die Reihen a und f) zeigen deutlich niedrigere Werte. Das liegt daran, dass sie weniger Gewinnlinien bieten. Ein Randfeld kann höchstens Teil einer horizontalen und einer diagonalen Viererreihe sein, weshalb dort oft Werte von 3 oder 4 zu finden sind.

Strategische Bedeutung:

Die zentralen Felder sind besonders wichtig, weil sie viele Optionen bieten, um eine Viererreihe zu vervollständigen. Deshalb ist es entscheidend, die mittleren Spalten – vor allem Spalte 4 sowie die angrenzenden Spalten 3 und 5 – im Auge zu behalten. Diese Kontrolle kann den Unterschied im Spiel ausmachen. Im Gegensatz dazu haben die Randfelder we-

niger Einfluss auf den Verlauf des Spiels. Sie werden oft hauptsächlich defensiv genutzt oder um den Gegner zu bestimmten Zügen zu zwingen.

5.3 Entwicklung von Algorithmen

In diesem Abschnitt werden verschiedene Algorithmen vorgestellt, die in unserem Projekt Anwendung finden können. Jeder Algorithmus wird zunächst in seinen Grundzügen erklärt, gefolgt von einer kritischen Bewertung zur Realisierung.

5.3.1 MinMax-Algorithmus

Die Minimax-Strategie ist ein grundlegender Algorithmus zur Entscheidungsfindung in Nullsummenspielen wie "Vier Gewinnt". Dieser Ansatz wird verwendet, um optimale Züge für einen Spieler zu finden, indem sowohl die eigenen Möglichkeiten als auch die möglichen Gegenreaktionen des Gegners analysiert werden. In diesem Kapitel betrachten wir die Funktionsweise der Minimax-Strategie im Kontext von "Vier Gewinnt" und zeigen, wie sie das Spielverhalten verbessern kann.

Grundprinzip der Minimax-Strategie

Der Minimax-Algorithmus funktioniert nach dem Prinzip der Maximierung des eigenen Nutzens und der Minimierung des Nutzens des Gegners. In einem Spielbaum, der alle möglichen Züge und Gegenreaktionen darstellt, sucht der Algorithmus nach dem optimalen Zug, indem er die Ergebnisse aller möglichen Züge bis zu einer bestimmten Tiefe bewertet.

In der Spieltheorie bezeichnet ein Spielbaum einen azyklischen, gerichteten Graphen, in dem die Knoten die verschiedenen Stellungen eines Spiels repräsentieren. Dabei kann dieselbe Stellung an mehreren Knoten des Spielbaums auftreten. Jeder Knoten kennzeichnet zudem, welcher Spieler in dieser Position am Zug ist. Der Spieler hat die Möglichkeit, eine der verfügbaren Aktionen auszuwählen, die durch die Kanten des Graphen dargestellt werden.

Jede Spielstellung im Baum ist entweder eine Endstellung oder besitzt eine Menge von Nachfolgestellungen, die durch die möglichen Züge im nächsten Halbzug des Spiels erreicht

werden können.

Die Bewertung erfolgt durch eine sogenannte Heuristik, die den Wert eines Spielzustands quantifiziert. In "Vier Gewinnt" könnte eine einfache Heuristik beispielsweise folgende Aspekte bewerten:

- Gewonnene Spiele: Ein Zustand, in dem ein Spieler vier Steine in einer Reihe hat, hat den höchsten Wert.
- Blockierte Gewinnchancen: Züge, die verhindern, dass der Gegner vier in einer Reihe erreicht, sind besonders wertvoll.
- Teilweise vervollständigte Reihen: Zustände, in denen drei oder zwei verbundene Steine existieren, sind wertvoller als isolierte Steine.

Der Algorithmus berechnet mögliche Züge und bewertet sie rückwärts ausgehend von den möglichen Endzuständen.

Maximierer und Minimierer

In "Vier Gewinnt" gibt es zwei Spieler:

- Der Maximierer versucht, den eigenen Nutzen zu maximieren (z. B. Spieler 1).
- Der Minimierer versucht, den Nutzen des Gegners zu minimieren (z. B. Spieler 2).

Bei jedem Zug wechselt die Rolle zwischen Maximierer und Minimierer. Der Algorithmus wechselt daher bei jedem Ebenenwechsel zwischen der Maximierung und Minimierung der Heuristikwerte.

Beispiel: Minimax-Suchbaum

Im Folgenden wird ein einfacher Suchbaum dargestellt, der die Funktionsweise des Minimax-Algorithmus illustriert. Der Baum hat eine Tiefe von 2 (eine Max-Ebene und eine Min-Ebene).

Suchbaum

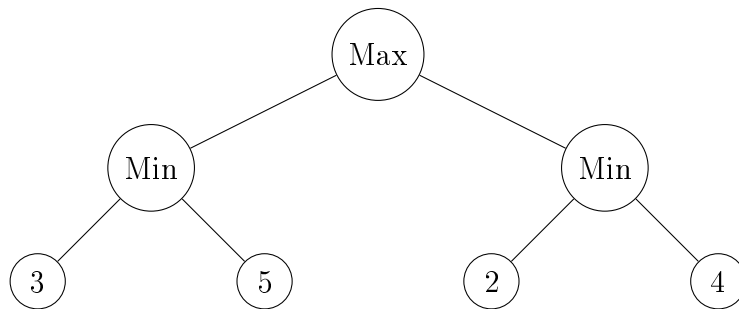


Abbildung 5.5: Ein einfacher Minimax-Suchbaum mit einer Tiefe von 2.

Erklärung des Suchbaums

Der Suchbaum in Abbildung 5.5 zeigt die Schritte des Minimax-Algorithmus:

1. **Blattebene:** Die unterste Ebene enthält die Bewertungen der möglichen Endzustände aus Sicht des Max-Spielers. Diese Werte könnten durch eine Bewertungsfunktion ermittelt worden sein.
 - Die Knoten haben die Werte 3, 5, 2 und 4.
2. **Min-Ebene:** Der Min-Spieler versucht, die Bewertung zu minimieren. Für jeden Min-Knoten wird der niedrigste Wert seiner Kindknoten ausgewählt:
 - Der linke Min-Knoten wählt $\min(3, 5) = 3$.
 - Der rechte Min-Knoten wählt $\min(2, 4) = 2$.
3. **Max-Ebene:** Der Max-Spieler versucht, die Bewertung zu maximieren. Der Max-Knoten wählt den höchsten Wert aus den zurückgegebenen Bewertungen der Min-Knoten:
 - Der Max-Knoten wählt $\max(3, 2) = 3$.

Optimale Entscheidung

Der optimale Zug für den Max-Spieler ist derjenige, der zu einer Bewertung von **3** führt.

Zusammenfassung

Der Minimax-Algorithmus analysiert den Baum von unten nach oben:

- Min-Knoten repräsentieren die Entscheidungen des Gegners (Minimierer).
- Max-Knoten repräsentieren die Entscheidungen des Spielers (Maximierer).

Durch diese systematische Analyse findet der Algorithmus den besten Zug für den Maximierer unter der Annahme, dass der Minimierer optimal spielt.

5.3.1 AlphaBeta-Algorithmus

Der Alpha-Beta-Algorithmus ist eine Optimierung des Minimax-Algorithmus, die dessen Effizienz erheblich steigert. Durch das gezielte Verwerfen von Spielzügen, die für die Entscheidung irrelevant sind, reduziert der Alpha-Beta-Algorithmus die Anzahl der bewerteten Knoten im Spielbaum. Diese Technik ist besonders wertvoll bei ressourcenbeschränkten Systemen wie einem LEGO Spike Roboter, der begrenzte Rechenleistung zur Verfügung hat.

Der Alpha-Beta-Algorithmus führt die gleichen Berechnungen wie der Minimax-Algorithmus durch, ergänzt jedoch zwei zusätzliche Parameter, Alpha und Beta, um unnötige Berechnungen zu vermeiden:

- **Alpha:** Der aktuelle maximale Wert, den der Maximierer sicher erreichen kann.
- **Beta:** Der aktuelle minimale Wert, den der Minimierer sicher erreichen kann.

Während der Traversierung des Spielbaums werden Äste abgeschnitten, die keine Auswirkungen auf die endgültige Entscheidung haben (*Pruning*). Dies geschieht, wenn:

- Ein Knoten einen Wert liefert, der schlechter ist als der bisher bekannte Alpha-Wert für den Maximierer.
- Ein Knoten einen Wert liefert, der schlechter ist als der bisher bekannte Beta-Wert für den Minimierer.

Die Anwendung des Alpha-Beta-Algorithmus auf "Vier Gewinnt" bietet mehrere Vorteile:

- **Effizienz:** Der Algorithmus reduziert die Anzahl der Knoten, die bewertet werden müssen, erheblich. Dies ermöglicht die Analyse tieferer Spielbäume mit derselben Rechenleistung.
- **Flexibilität:** Der Algorithmus kann leicht an die spezifische Bewertungsfunktion von "Vier Gewinnt" angepasst werden, z. B. zur Bewertung von Reihen, Spalten und Diagonalen.
- **Optimierung für begrenzte Ressourcen:** Ein LEGO Spike Roboter hat begrenzte Rechen- und Speicherkapazitäten. Der Alpha-Beta-Algorithmus ermöglicht es, in Echtzeit Züge zu berechnen, ohne den Roboter zu überlasten.

Beispiel: Alpha-Beta-Suchbaum

Im Folgenden wird ein einfacher Suchbaum dargestellt, der die Funktionsweise des Alpha-Beta-Algorithmus illustriert. Der Baum zeigt, wie bestimmte Zweige (*Pruning*) abgeschnitten werden, um die Effizienz zu erhöhen.

Suchbaum mit Alpha-Beta-Pruning

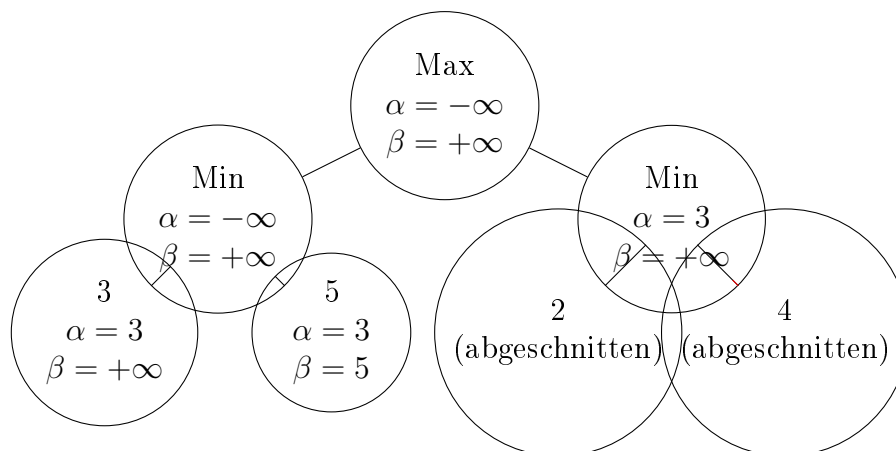


Abbildung 5.6: Ein Alpha-Beta-Suchbaum mit Pruning. Zweige werden abgeschnitten, wenn sie die Grenzen von Alpha und Beta verletzen.

Erklärung des Suchbaums

Der Suchbaum in Abbildung 5.6 zeigt die Schritte des Alpha-Beta-Algorithmus:

1. **Initialisierung:** Die Alpha- und Beta-Werte werden am Wurzelknoten initialisiert:

- $\alpha = -\infty$: Das beste Ergebnis, das der Maximierer bisher erreicht hat.
- $\beta = +\infty$: Das schlechteste Ergebnis, das der Minimierer dem Maximierer erlauben würde.

2. **Linker Teilbaum:**

- Der Maximierer überprüft den linken Teilbaum, in dem der Minimierer aktiv ist.
- Der Min-Knoten prüft seine Kindknoten:
 - Der erste Kindknoten liefert den Wert 3. Der Alpha-Wert wird aktualisiert: $\alpha = 3$.
 - Der zweite Kindknoten liefert den Wert 5. Da 5 größer ist, bleibt $\alpha = 3$ bestehen.

3. **Rechter Teilbaum (Pruning):**

- Der Maximierer prüft den rechten Teilbaum. Der Min-Knoten beginnt mit der Analyse.
- Bereits beim ersten Kindknoten erkennt der Algorithmus, dass dessen Wert (2) kleiner ist als $\alpha = 3$.
- Alle verbleibenden Kindknoten des rechten Teilbaums werden abgeschnitten (*Pruning*), da sie keinen besseren Wert liefern können.

Vorteile des Alpha-Beta-Algorithmus

Durch das Alpha-Beta-Pruning werden unnötige Berechnungen vermieden:

- Der Algorithmus durchsucht nur jene Zweige, die potenziell zu besseren Ergebnissen führen.
- In diesem Beispiel werden zwei Knoten (Wert 2 und 4 im rechten Teilbaum) nicht berechnet.
- Dies erhöht die Effizienz und ermöglicht es, tiefere Suchbäume mit derselben Rechenleistung zu analysieren.

6 Anwendung der Spieltheorie auf Vier Gewinnt

In diesem Kapitel werden die spieltheoretischen Aspekte von Vier Gewinnt, einschließlich der Analyse von Spielsituationen, der Berechnung von Gewinnwahrscheinlichkeiten und der Entwicklung von Spielstrategien untersucht.

6.1 Analyse von Spielsituationen

Für das Verständnis der Spieldynamik und die Entwicklung effektiver Strategien, ist die Analyse der Spielsituation entscheidend. Vier Gewinnt, ist ein zwei Personen Spiel mit vollständiger Information. Das bedeutet, alle Spieler zu jeder Zeit den gesamten Zustand des Spiels kennen [Rui+09].

Der Bestimmungssatz von Ernst Zermelo besagt, dass sich jedes kombinatorische Spiel in eine der folgenden Kategorien einordnen lässt.

1. Der beginnende Spieler hat eine dominante Strategie. Wenn das Spiel optimal, ohne Fehler verläuft, gewinnt er die Partie.
2. Der nachziehende Spieler 2 hat eine dominante Strategie. Wenn das Spiel optimal verläuft, gewinnt dieser.
3. Keiner der beiden Parteien hat eine dominante Strategie. Wenn beide Spieler optimal spielen, endet das Spiel ohne einen Sieger. Sobald ein Spieler einen Fehler macht, gewinnt der andere Spieler das Spiel [Mül11].

Aus der Analyse geht hervor, dass bestimmte Situationen von besonderer Bedeutung für einen Sieg sind. Das bilden einer horizontalen Viererreihe sind schwer zu erreichen. Umso

wichtige ist es einen Sieg die Spalten 1-5 zu kontrollieren.

6.2 Berechnung von Gewinnwahrscheinlichkeiten

Aufgrund der Komplexität des Spiels stellt die Berechnung der Gewinnwahrscheinlichkeit in Anwendung eine Herausforderung dar. Das Spielraster besteht aus sieben Spalten und sechs Zeilen, dadurch entsteht eine enorm große Anzahl an möglichen Zuständen. Was die Wahrscheinlichkeitsberechnung für den Gewinn des Spiels so komplex macht. Moderne Ansätze zur Berechnung der Gewinnwahrscheinlichkeit greifen auf KI-Technik zurück. Die exakte Berechnung für alle möglichen Spielsituationen bleibt aufgrund der Komplexität von Vier Gewinnt eine Herausforderung [Rui+09].

6.3 Entwicklung von Spielstrategien

Um eine spielstarke Strategie für "Vier Gewinnt" mit dem Alpha-Beta-Algorithmus zu entwickeln und später zu programmieren, müssen mehrere Schritte beachtet werden. Hier wird der gesamte Prozess erläutert, von der Spielrepräsentation über die Bewertung von Stellungen bis hin zur Integration des Alpha-Beta-Algorithmus.

1. Spielfeld als Datenstruktur

Zunächst muss das Spielfeld von "Vier Gewinnt" in einer Form dargestellt werden, die von einem Algorithmus verarbeitet werden kann:

Verwende ein 2D-Array (6 Zeilen \times 7 Spalten), das den Zustand des Spielfeldes beschreibt.

- 0: Leeres Feld
- 1: Spielstein von Spieler 1
- -1: Spielstein von Spieler 2

Das Spielfeld kann folgendermaßen dargestellt werden:

$$\text{Spielfeld} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

2. Zugmöglichkeiten berechnen

Da Spielsteine nur in den unteren freien Reihen platziert werden können, muss eine Funktion implementiert werden, die gültige Züge berechnet:

Listing 6.1: Funktion zur Berechnung gültiger Züge

```
1 def valid_moves(board):  
2     return [col for col in range(7) if board[0][col] == 0]
```

3. Bewertung der Spielstellung

Eine Bewertungsfunktion ist entscheidend für die Strategie. Sie schätzt den Wert einer Stellung ein und liefert eine Zahl:

- **Gewinn für Spieler:** Sehr hoher Wert, z. B. +1000.
- **Verlust für Gegner:** Sehr niedriger Wert, z. B. −1000.
- **Potenzielle Verbindungen:** Punkte für 2er- und 3er-Reihen, die noch zu einem Sieg führen können.
- **Blockieren von Gegnerzügen:** Zusätzliche Punkte für Züge, die den Gegner daran hindern, 4 in eine Reihe zu bilden.

4. Alpha-Beta-Algorithmus implementieren

Der Alpha-Beta-Algorithmus wird verwendet, um den besten Zug zu berechnen, indem der Suchbaum effizient durchsucht wird. Dieses Beispiel zeigt den Kern des Algorithmus – die Auswahl des besten Zugs basierend auf einer vorgegebenen Spielbrett-Situation (board) und einer Suchtiefe (depth).

Listing 6.2: Alpha-Beta Algorithmus - Kurzer Überblick

```
1  # Alpha-Beta Algorithmus in Aktion
2  best_move = alpha_beta(board, depth=4, alpha=-float('inf'), beta=float('inf'), maximizingPlayer=True)
3  print(f"Bester Zug: Spalte {best_move}")
```

5. Entscheidung für den besten Zug

Im fünften Schritt wird der beste Zug für den aktuellen Spieler ermittelt. Dazu werden alle möglichen Züge analysiert, indem sie simuliert und mit dem Alpha-Beta-Algorithmus bewertet werden. Der Zug mit der besten Bewertung (je nach Spieler maximierend oder minimierend) wird ausgewählt. Dieser Schritt stellt die Grundlage für die Entscheidungsfindung dar und bildet den Abschluss der Spielbaum-Analyse.

Listing 6.3: Entscheidung für den besten Zug - Überblick

```
1  # Grober Aufbau der Funktion zur Zugentscheidung
2  def best_move(board, depth, player):
3      best_value = float('-inf') if player == 1 else float('inf')
4      best_column = None
5
6      for col in valid_moves(board):
7          move_value = alpha_beta(apply_move(board, col, player), depth - 1, -float('inf'), float('inf'), False)
8          if (player == 1 and move_value > best_value) or (player == -1 and move_value < best_value):
9              best_value = move_value
10             best_column = col
11
12     return best_column
```

- Die Funktion `best_move` ist darauf ausgelegt, den besten Zug für einen Spieler (`player`) zu bestimmen.

- `valid_moves(board)` gibt alle gültigen Spalten zurück, in die ein Stein gesetzt werden kann.
- `apply_move(board, col, player)` simuliert das Setzen eines Steins in eine bestimmte Spalte.
- Der *alpha-beta*-Algorithmus wird auf das simulierte Spielfeld angewendet, um die Bewertung des Zugs zu berechnen.
- Der Spieler wählt den Zug mit der höchsten Bewertung (für Maximierer) oder der niedrigsten Bewertung (für Minimierer).

5. Endzustände erkennen

Eine Funktion wird benötigt, um festzustellen, ob das Spiel vorbei ist:

Kriterien für Endzustände:

- Einer der Spieler hat 4 in einer Reihe.
- Das Spielfeld ist voll.

Listing 6.4: Erkennung des Endzustands

```
1 def is_terminal(board):  
2     return has_winner(board) or all(board[0][col] != 0 for col in range(7))
```

7. Spielstrategie optimieren

Suchtiefe anpassen

Je nach Rechenleistung kann die Tiefe des Spielbaums variiert werden.

Zeitlimit setzen

Da der Algorithmus auf einem LEGO Spike Roboter läuft, ist die Rechenzeit begrenzt. Eine Tiefensuche kann eingesetzt werden, um innerhalb eines Zeitlimits die bestmögliche Tiefe zu erreichen.

6.4 Systementwurf

Es soll eine detaillierte und funktionsfähige Vorrichtung entworfen werden, die in der Lage ist, Spielsteine präzise und zuverlässig in die vorgesehenen Spalten des Spielständers einzuführen. Dazu muss der Mechanismus so gestaltet sein, dass er unterschiedliche Positionen des Spielständers ansteuern und die Spielsteine mit der nötigen Genauigkeit platzieren kann. Nach dem Einwurf des Spielsteins muss die Vorrichtung den Spielständer wieder für den nächsten Zug freigeben, was eine genaue Steuerung und Rückkehr in die Ausgangsposition erfordert. Die mechanische Konstruktion muss robust und wiederholgenau arbeiten, um einen flüssigen und störungsfreien Spielablauf zu gewährleisten.

6.4.1 Komponenten

Zur Verwendung stehen folgende Sensorik und Aktorik zur Verfügung:

- **großer LEGO® Technic Hub:** Der Hub ist das Steuerungselement des Spike Prime Systems. Er umfasst sechs Ein-/Ausgänge, welche den Anschluss von Peripheriegeräten, wie Sensorik und Aktorik, ermöglicht. Mit einem Micro-USB-Anschluss und Bluetooth wird die Kommunikation mit kompatiblen Endgeräten hergestellt. Der Hub besitzt ein integriertes MicroPython-Betriebssystem mit einem 100-MHz-Prozessor. Weitere Ausstattungen sind: Individuell anpassbaren Lichtmatrix (5x5) Aufzeigen von wichtigen Informationen und Statusmeldungen Tasten Ermöglichen eine einfache Navigation und Steuerung durch Menüs Lautsprecher 6-achsigen Kreiselnsensor

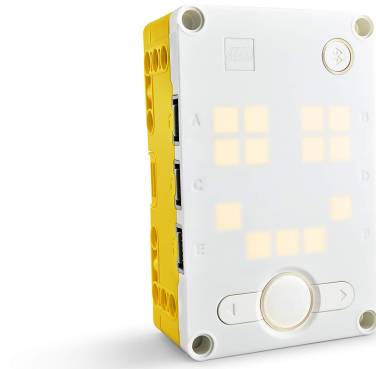


Abbildung 6.1: Lego Technic Hub

- **LEGO® Technic Farbsensor:** Dieser Sensor kann durch die Intensität des reflektierten Lichts, welche er durch den eingebauten Lichtring erzeugt, bis zu acht Farben erkennen und unterscheiden (erkennbare Farben: schwarz, blau, rot, weiß, braun, gelb, pink und grün). Er wird vor allem bei Anwendungen, wie Sortieren nach Farben, Linienverfolgung von farbigen Streifen und zum Ausführen von farbcodierten Befehlen eingesetzt. Um die Farberkennung optimal auf die entsprechende Umgebung anzupassen, wird das Umgebungslicht zuvor ausgewertet.



Abbildung 6.2: Lego Technic Farbsensor

- **LEGO® Technic Großer Winkelmotor** Für die Bewegungssteuerung wird dieser Winkelmotor mit hoher Drehkraft und präziser Steuerung verwendet. Dieser Motor ist dafür ausgelegt, um schwere und komplexe Konstruktionen exakt in Geschwindigkeit und Position verfahren zu können. Dies wird durch den internen Winkel- und Rotationssensor ermöglicht. Der Anwendungsbereich ist vielseitig, unter anderem wird dies als Antrieb von Rädern, Gelenke, Greifarme, Hebevorrichtungen und Drehscheibe angewandt. Vor allem für Aufgaben, welche eine genau und wiederholbare Aktion ausüben müssen.



Abbildung 6.3: Lego Technic Winkelmotor

6.4.2 Zusammenspiel von Soft- und Hardware

Der LEGO SPIKE Farbsensor eignet sich hervorragend zur Erkennung der Spielsteine in einem Vier-Gewinnt-Spiel.

6.5 Mechanische Konstruktion des Farbsensor-Systems

Das Farbsensor-System basiert auf einem zweiachsigen Antriebssystem, das eine präzise Bewegung sowohl in vertikaler als auch in horizontaler Richtung ermöglicht.

Die **vertikale Bewegung** wird durch eine Kettenkonstruktion realisiert, an der der Farbsensor befestigt ist. Ein LEGO SPIKE Motor treibt die Kette präzise an und sorgt dafür, dass der Sensor alle sechs Reihen des Spielfeldes abtasten kann.

Die **horizontale Bewegung** erfolgt über einen Wagen mit vier Rädern. Dieser Wagen wird durch einen zweiten Motor angetrieben, der mit Encoderwerten für eine präzise Positionierung gesteuert wird. Durch die horizontale Bewegung können alle sieben Spalten des Spielfeldes abgedeckt werden.

Die **Steuerung der Bewegungen** erfolgt durch eine koordinierte Ansteuerung beider Achsen, wodurch der Farbsensor schrittweise von Position zu Position bewegt wird. Optimierte Fahrwege minimieren die Scanzeit, und eine Kalibrierung der Positionen gewährleistet eine genaue Ausrichtung des Sensors.

Diese mechanische Konstruktion ermöglicht eine zuverlässige und präzise Abtastung aller 42 Spielfeldpositionen des Vier-Gewinnt-Spiels und stellt somit eine optimale Grundlage für die Umsetzung dar.

7 Zusammenfassung

Die spieltheoretische Analyse von "Vier Gewinnt" hat gezeigt, dass das Spiel trotz seiner scheinbaren Einfachheit eine hohe strategische Tiefe besitzt. Durch die Anwendung von Algorithmen wie Minimax und dessen Optimierung durch den Alpha-Beta-Algorithmus können optimale Entscheidungen getroffen werden, um entweder zu gewinnen oder ein Unentschieden zu erzwingen.

Der Alpha-Beta-Algorithmus ist hierbei besonders hervorzuheben, da er durch die effiziente Kürzung des Suchbaums eine präzise Analyse in kürzerer Zeit ermöglicht. Diese Eigenschaft ist für die praktische Umsetzung auf ressourcenbeschränkten Systemen wie dem LEGO SPIKE Roboter entscheidend. Der Algorithmus gewährleistet, dass der Roboter in der Lage ist, komplexe Spielsituationen in Echtzeit zu bewerten und optimale Züge zu berechnen.

Mit den gewonnenen Erkenntnissen aus der Spieltheorie und der mechanischen Konstruktion des Farbsensor-Systems ist nun der Grundstein für den zweiten Teil dieser Studienarbeit gelegt. In diesem Teil wird ein Programm entwickelt, das den Alpha-Beta-Algorithmus implementiert und mit der Hardware des LEGO SPIKE Roboters zusammenarbeitet.

Das Programm wird die Daten des Farbsensors nutzen, um den aktuellen Spielstand zu erkennen und auf dieser Basis optimale Spielzüge zu berechnen. Durch die präzise Steuerung der mechanischen Komponenten wird der Roboter in der Lage sein, seine Züge zuverlässig auszuführen. Diese Kombination aus theoretischen Grundlagen, Software und Hardware stellt eine vollständige Lösung dar, die sowohl den Anforderungen eines funktionierenden Vier-Gewinnt-Spielers als auch den didaktischen Zielen der Studienarbeit gerecht wird.

Literaturverzeichnis

- [50P25] 50PLUS. *Vier Gewinnt - Online Spiele*. Online; abgerufen am 1. Januar 2025. 50PLUS, 2025. URL: <https://www.50plus.de/spiele/vier-gewinnt.html>.
- [Ein14] Johanna Einsiedler. „Spieltheorie: Bimatrixspiele und deren praktische Anwendung“. Fachbereichsarbeit. St. Pölten: Mary Ward Privatgymnasium und Oberstufenrealgymnasium St. Pölten, 2014. URL: https://www.oemg.ac.at/Mathe-Brief/fba2014/Spieltheorie_Einsiedler.pdf.
- [Fas24] FasterCapital. *Den Code knacken: Rückwärtsinduktion und Payoff-Matrix-Analyse*. FasterCapital. 2024. URL: <https://fastercapital.com/de/inhalt/Den-Code-knacken--Rueckwaertsinduktion-und-Payoff-Matrix-Analyse.html> (besucht am 05.01.2025).
- [Gam25] Gameorama. *4 Gewinnt*. Spielmuseum Timeline. Gameorama, 2025. URL: <https://www.gameorama.ch/de/museum/spielmuseum/timeline/gewinnt> (besucht am 01.01.2025).
- [Has20] SA Hasbro. *Das Originale 4Gewinnt Anleitung*. Hasbro Gaming, 2020.
- [Hol19] Manfred J. Holler. *Einführung in die Spieltheorie*. ger. Hrsg. von Gerhard Illing und Stefan Napel. 8. Auflage. Lehrbuch. Berlin: Springer Gabler, 2019. ISBN: 978-3-642-31962-4. URL: http://deposit.dnb.de/cgi-bin/dokserv?id=5314105&prov=M&dok_var=1&dok_ext=htm.
- [Mül11] Markus Müller. „Seminararbeitsaus dem Fach Mathematik“. de. In: (2011). URL: https://www.edu.sot.tum.de/fileadmin/w00bed/edu/Schuelerkonferenz/Seminararbeiten_2012/mueller_markus_2012.pdf (besucht am 18.12.2024).
- [Rui+09] Benjamin Ruile u. a. *Vier Gewinnt*. Vorlesung Spieltheorie. Ludwig-Maximilians-Universität München, 2009. URL: <https://www.mathematik.uni-muenchen.de/~spielth/artikel/VierGewinnt.pdf>.
- [Sie] Dr. Markus Siepermann. *Normalform*. SpringerGabler. URL: <https://wirtschaftslexikon.gabler.de/definition/normalform-39481> (besucht am 08.01.2025).

- [Thi12] Markus Thill. „Reinforcement Learning mit N-Tupel-Systemen für Vier Gewinnt“. Bachelor’s Thesis. Gummersbach: Technische Hochschule Köln, Juni 2012. URL: <https://www.gm.th-koeln.de/~konen/research/PaperPDF/Bachelorarbeit%20Thill-2012.pdf>.
- [Wik25] Wikipedia-Autoren. *Vier gewinnt*. Online; abgerufen am 1. Januar 2025. Wikipedia, Die freie Enzyklopädie, 2025. URL: https://de.wikipedia.org/wiki/Vier_gewinnt.
- [YGJ22] Justin Yokota, Dan Garcia Ed. und James James Demmel Ed. *High Efficiency Computation of Game Tree Exploration in Connect 4*. Techn. Ber. UCB/EECS-2022-219. Electrical Engineering und Computer Sciences, University of California, Berkeley, Aug. 2022. URL: <https://www2.eecs.berkeley.edu/Pubs/TechRpts/2022/EECS-2022-219.pdf>.

Abbildungsverzeichnis

| | | |
|-----|---|----|
| 2.1 | Mögliche Spielausgänge | 3 |
| 2.2 | Sogo von Ravensburger [abebooks_image] | 4 |
| 2.3 | Payoff-Matrix für Vier gewinnt | 5 |
| 4.1 | Spielbaum nach der 2. Tiefe | 13 |
| 4.2 | Spielbaum nach dem 4. Zug. Quelle: [vandewiele2017] | 14 |
| 5.1 | ROT gewinnt bei optimalen Spiel | 17 |
| 5.2 | Remis bei optimalen Spiel | 18 |
| 5.3 | ROT verliert bei optimalen Spiel | 18 |
| 5.4 | Bewertungstabelle: mögliche Anzahl an Vierer-Reihen | 19 |
| 5.5 | Ein einfacher Minimax-Suchbaum mit einer Tiefe von 2. | 22 |
| 5.6 | Ein Alpha-Beta-Suchbaum mit Pruning. Zweige werden abgeschnitten, wenn sie die Grenzen von Alpha und Beta verletzen. | 24 |
| 6.1 | Lego Technic Hub | 33 |
| 6.2 | Lego Technic Farbsensor | 33 |
| 6.3 | Lego Technic Winkelmotor | 34 |

Tabellenverzeichnis