Benutzerhandbuch

Softwaretechnik-Projekt SoSe2024

Thema

Plattform zum Vergleich von Spiele-KIs: **KIMaster**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Änderungshistorie** | | | | |
| **Version** | **Datum** | **Kapitel** | **Änderung** | **Name** |
| 0.1 | 04.07.2024 | Alle | Anlegen und Füllen | Justine Buß |
| 0.2 | 20.07.2024 | 4 | Schreiben | Thorben Jones |
| 0.3 | 21.07.2024 | 4 | Nachbessern | Alexander Roos |
| 0.4 | 21.07.2024 | 3 | Schreiben | Omar Karkotli |
| 0.5 | 23.07.2024 | 3 | Schreiben | Omar Karkotli |
| 0.6 | 24.07.2024 | 3, 6 | Nachbessern, Schreiben | Justine Buß |
| 1.0 | 24.07.2024 | 3 | Bilder einfügen | Omar Karkotli |

|  |  |  |
| --- | --- | --- |
| Herausgeber | [Technische Hochschule Mittelhessen – FB06 Mathematik, Naturwissenschaften und Informatik](https://www.thm.de/mni/) | |
| Dateiname | Benutzerhandbuch\_ver\_1.0 | |
| Dokumentenbezeichnung | Benutzerhandbuch: Plattform zum Vergleich von Spiele-KIs: KIMaster | |
| Version | 1.0 | |
| Stand | Donnerstag, 25. Juli 2024 | |
| Status | Fertiggestellt | |
| Autoren | Justine Buß, Thorben Jones, Alexander Roos, Maximilian Bachmann, Omar Karkotli, Sven Roman Reinhard, Pascal Waldschmidt | |
| Freigegeben von |  |  |
| Ansprechpartner | Justine Buß | [justine.buss@mni.thm.de](mailto:justine.buss@mni.thm.de) |
|  | Thorben Jones | [thorben.jones@mni.thm.de](mailto:thorben.jones@mni.thm.de) |
| Kurzinfo | „Technische Hochschule Mittelhessen Softwaretechnik-Projekt. Benutzerhandbuch für KIMaster.“ | |

Inhaltsverzeichnis

[1 Einführung 8](#_Toc172802054)

[1.1 Kurzübersicht der Software 8](#_Toc172802055)

[1.2 Zielgruppen 8](#_Toc172802056)

[2 Einrichtung 9](#_Toc172802057)

[2.1 Systemanforderungen 9](#_Toc172802058)

[3 Benutzeranleitung für die Webseite 11](#_Toc172802059)

[3.1 Webseite aufrufen 11](#_Toc172802060)

[3.2 Startseite 12](#_Toc172802061)

[3.2.1 Navigationsleiste 12](#_Toc172802062)

[3.2.2 Begrüßung 13](#_Toc172802063)

[3.2.3 Spielauswahl 13](#_Toc172802064)

[3.2.4 Eingabe des Lobbyschlüssels 14](#_Toc172802065)

[3.2.5 Footer 15](#_Toc172802066)

[3.3 Warteraum 17](#_Toc172802067)

[3.4 Lobby-Seite 18](#_Toc172802068)

[3.4.1 Spielekonfiguration 19](#_Toc172802069)

[3.4.2 Status und Lobby-Schlüssel 20](#_Toc172802070)

[3.5 Spielseite 22](#_Toc172802071)

[3.5.1 Spielbrett 24](#_Toc172802072)

[3.5.4 Fertiges Spiel 29](#_Toc172802073)

[3.5.5 Zeitleiste 30](#_Toc172802074)

[3.5.6 Fehleranalyse 31](#_Toc172802075)

[3.5.7 Mini-Fenster 32](#_Toc172802076)

[4 Benutzeranleitung für die KIMaster-Schnittstelle 33](#_Toc172802077)

[4.1 Überblick über die Schnittstelle (Python Beispiel) 33](#_Toc172802078)

[4.2 Kommandos 34](#_Toc172802079)

[4.3 Voraussetzungen 35](#_Toc172802080)

[4.4 Einrichtung 35](#_Toc172802081)

[4.5 Nutzung der Hauptfunktionen 37](#_Toc172802082)

[5 Häufige Probleme und Lösungen 39](#_Toc172802083)

[5.1 Meine Verbindung bricht ständig ab, wenn ich etwas empfange? 39](#_Toc172802084)

[5.2 Ich kann mich nicht mit dem THM-Server verbinden? 39](#_Toc172802085)

[5.3 Ich erhalte keine Antworten trotz While True loop? 40](#_Toc172802086)

[6 Literaturverzeichnis 41](#_Toc172802087)

Abbildungsverzeichnis

[Abbildung 1: Verbindungsherstellung THM VPN 10](#_Toc172802088)

[Abbildung 2: Verbindung hergestellt THM VPN 10](#_Toc172802089)

[Abbildung 3: URL-Eingabe in Suchleiste des Browsers 11](#_Toc172802090)

[Abbildung 4: Startseite 12](#_Toc172802091)

[Abbildung 5: Navigationsleiste 12](#_Toc172802092)

[Abbildung 6: Begrüßung 13](#_Toc172802093)

[Abbildung 7: Spieleauswahl 13](#_Toc172802094)

[Abbildung 8: Spieleauswahl Hovering 14](#_Toc172802095)

[Abbildung 9: Lobby beitreten 14](#_Toc172802096)

[Abbildung 10: Lobby beitreten - Fehleingabe 14](#_Toc172802097)

[Abbildung 11: Footer 15](#_Toc172802098)

[Abbildung 12: Datenschutzerklärung 15](#_Toc172802099)

[Abbildung 13: Impressum THM 16](#_Toc172802100)

[Abbildung 14: Warteraum: belegte Position Spieler 1 17](#_Toc172802101)

[Abbildung 15: Lobby-Seite 18](#_Toc172802102)

[Abbildung 16: Spielerkonfiguration 19](#_Toc172802103)

[Abbildung 17: Spielmoduskonfiguration 19](#_Toc172802104)

[Abbildung 18: Schwierigkeitskonfiguration 19](#_Toc172802105)

[Abbildung 19: Spielelobbywechsel 20](#_Toc172802106)

[Abbildung 20: Lobby-Schlüssel & QR-Code 20](#_Toc172802107)

[Abbildung 21: QR-Code 20](#_Toc172802108)

[Abbildung 22: Positionsanzeige 21](#_Toc172802109)

[Abbildung 23: Lobby-Aktionen 21](#_Toc172802110)

[Abbildung 24: Spieleseite) 22](#_Toc172802111)

[Abbildung 25: Spielregeln 23](#_Toc172802112)

[Abbildung 26: Zurückgenommener Zug 24](#_Toc172802113)

[Abbildung 27: Valide Züge 24](#_Toc172802114)

[Abbildung 28: Maus-Hoover 25](#_Toc172802115)

[Abbildung 29: Ausgeführter Zug 25](#_Toc172802116)

[Abbildung 30: Dame-Brett 26](#_Toc172802117)

[Abbildung 31: Mögliche Züge anzeigen 26](#_Toc172802118)

[Abbildung 32: Zug ausgeführt (Dame) 27](#_Toc172802119)

[Abbildung 33: Nim-Brett 28](#_Toc172802120)

[Abbildung 34: Ausgewählte Reihe 28](#_Toc172802121)

[Abbildung 35: Zug bestätigt 29](#_Toc172802122)

[Abbildung 36: Spiel vorbei 29](#_Toc172802123)

[Abbildung 37: Spielende zusätzliche Funktionen 30](#_Toc172802124)

[Abbildung 38: Zeitleiste - vorherig 30](#_Toc172802125)

[Abbildung 39: Fehleranalyse 31](#_Toc172802126)

[Abbildung 40: Mini-Fenster 32](#_Toc172802127)

[Abbildung 41: Ordnerstruktur Python Schnittstelle 34](#_Toc172802128)

[Abbildung 42: Install-Befehl Bibliotheken 35](#_Toc172802129)

[Abbildung 43: Zu verbindende URIs 35](#_Toc172802130)

[Abbildung 44: connect() 36](#_Toc172802131)

[Abbildung 45: main() 36](#_Toc172802132)

[Abbildung 46: send\_cmd() 37](#_Toc172802133)

[Abbildung 47: receive() 38](#_Toc172802134)

[Abbildung 48: Hauptprogramm 38](#_Toc172802135)

[Abbildung 49: Formatüberprüfung 39](#_Toc172802136)

[Abbildung 50: receive\_handler() ohne sleep 40](#_Toc172802137)

[Abbildung 51: send\_handler() mit sleep 40](#_Toc172802138)

# 1 Einführung

Dieses Benutzerhandbuch dient als Leitfaden für die Navigation und Nutzung der Plattform KIMaster. Es richtet sich sowohl an Nutzer, die über die Webseite spielen möchten, als auch an solche, die komplexere Anwendungsfälle, wie die Anbindung an die Schnittstelle und die Implementierung eigener Künstlicher Intelligenzen (KI), realisieren wollen.

## 1.1 Kurzübersicht der Software

Die Software stellt eine vielseitige Plattform bereit, auf der Nutzer in verschiedenen Spielen gegeneinander antreten können. Ein Alleinstellungsmerkmal ist die Möglichkeit, gegen vortrainierte KIs zu spielen, die für jedes Spiel verfügbar sind. Die Plattform kann über eine Webseite genutzt werden oder über ein Terminal, wenn die Anbindung an die bereitgestellte Schnittstelle erfolgt.

Ein zusätzliches Feature der Software ist die Option, eigene KI-Implementierungen zu integrieren. Entwickler können ihre KIs gegen andere Spieler, deren KIs oder vortrainierte KIs testen. Dies eröffnet vielfältige Möglichkeiten für Entwicklungen im Bereich der KI.

## 1.2 Zielgruppen

Dieses Benutzerhandbuch richtet sich an eine vielfältige Nutzergruppe:

* **Allgemeiner Spieler**: Personen, die auf der Webseite spielen und gegen andere Nutzer oder vortrainierte KIs antreten möchten.
* **Studierende**: Nutzer, die erste Erfahrungen mit KI sammeln oder spezielle KI-Module belegen und ihre eigenen KIs entwickeln möchten.
* **Dozenten:** Lehrkräfte, die die Plattform für Lehr- und Forschungszwecke im Bereich KI verwenden.
* **Entwickler**: Nutzer, die die Plattform über die Schnittstelle nutzen und eigene KI-Implementierungen integrieren möchten.

# 2 Einrichtung

## 2.1 Systemanforderungen

**1. Zugriff über die Webseite**

Für den Zugriff über die Webseite sind folgende Anforderungen zu erfüllen:

* **Internetfähiger Browser**: Ein moderner Browser wie [Google Chrome](https://www.google.com/intl/de_de/chrome/), [Mozilla Firefox](https://www.mozilla.org/de/firefox/), [Apple Safari](https://www.apple.com/de/safari/) oder [Microsoft Edge](https://www.microsoft.com/de-de/edge?form=MA13FJ) ist erforderlich.
* **Netzwerkverbindung**: Der Zugriff muss innerhalb des Netzwerks der [Technischen Hochschule Mittelhessen](https://www.thm.de/site/) erfolgen. Dies kann entweder direkt vor Ort oder durch Nutzung eines [VPN-Dienstes der THM](https://www.thm.de/its/campusnetz/vpn/ciscoanyconnect.html) geschehen.

**2. Externe Anbindung**

Für die externe Anbindung an die Systeme der THM sind folgende Voraussetzungen notwendig:

* **Netzwerkverbindung der THM**: Der Zugriff muss über das Netzwerk der Technischen Hochschule Mittelhessen erfolgen, was entweder direkt vor Ort oder durch die Nutzung eines [VPN-Dienstes der THM](https://www.thm.de/its/campusnetz/vpn/ciscoanyconnect.html) möglich ist.
* **Internetverbindung**: Eine stabile Internetverbindung ist erforderlich.
* **Programmierumgebung**: Eine Entwicklungsumgebung (IDE) oder ein Editor, der die gewählte Programmiersprache unterstützt, muss installiert sein.
* **WebSocket-Verbindungen**: Alle Verbindungen zu den Systemen der THM werden über [WebSocket](https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API)-Verbindungen realisiert. Stellen Sie sicher, dass Ihre Programmierumgebung und die verwendete Sprache die Einrichtung und Nutzung von WebSocket-Verbindungen unterstützen.

**Zusätzliche Hinweise**

* Stellen Sie sicher, dass alle Sicherheitsupdates und Patches für den verwendeten Browser und die Programmierumgebung installiert sind.
* Für die Nutzung des VPN-Dienstes der THM folgen Sie bitte den Anweisungen auf der offiziellen Webseite der THM: <https://www.thm.de/its/campusnetz/vpn/ciscoanyconnect.html> oder kontaktieren Sie den IT-Support bei möglichen Problemen.

Ein Bild, das Text, Screenshot, Software, Schrift enthält.

Automatisch generierte Beschreibung

Abbildung 1: Verbindungsherstellung THM VPN

Ein Bild, das Text, Screenshot, Schrift, Software enthält.

Automatisch generierte Beschreibung

Abbildung 2: Verbindung hergestellt THM VPN

* Überprüfen Sie, ob zusätzliche Bibliotheken oder Frameworks für die Unterstützung von WebSocket-Verbindungen in Ihrer Programmierumgebung erforderlich sind, und installieren Sie diese entsprechend.
  + [Python](https://www.python.org/): websockets, socket.io, tornado
  + [Java](https://www.java.com/de/): Java.WebSocket, Tyrus, Spring WebSocket
  + [C#](https://learn.microsoft.com/de-de/dotnet/csharp/tour-of-csharp/): SignalR, WebSocketSharp
  + [Rust](https://www.rust-lang.org/): tokio-tungstenite, async-tungstenite
  + [Kotlin](https://kotlinlang.org/): ktor, OkHttp
* Wir bieten auf unserem [GitHub-Repsoitory](https://github.com/12ghostrider21/KIMaster/tree/main/External/Python) ein Verbindungsbeispiel mit Python an.

# 3 Benutzeranleitung für die Webseite

## 3.1 Webseite aufrufen

1. Stellen Sie sicher, dass Sie die [Systemanforderungen](#_2.1_Systemanforderungen) erfüllen.
2. Öffnen Sie einen Webbrowser Ihrer Wahl.
   * Google Chrome
   * Apple Safari
   * Mozilla Firefox
   * Microsoft Edge
   * …
3. Geben Sie die URL kimaster.mni.thm.de in die Adressleiste Ihres Browsers ein.



Abbildung 3: URL-Eingabe in Suchleiste des Browsers

1. Bestätigen Sie Ihre Eingabe durch Drücken der Eingabetaste (Enter).

## 3.2 Startseite



Abbildung 4: Startseite

Die Startseite von KIMaster, kurz KIM genannt, ist übersichtlich und benutzerfreundlich gestaltet. Das Design ist minimalistisch und fokussiert sich darauf, dem Benutzer eine einfache Navigation und eine klare Struktur zu bieten.

Die Seite verwendet ein schlichtes Farbschema mit weißen Hintergründen und grünen Akzenten, die die Markenfarben der THM widerspiegeln.

Sie ist so gestaltet, dass sie auf verschiedenen Geräten gut aussieht und funktioniert. Ob auf einem Desktop-Computer, Tablet oder Smartphone, die Elemente der Seite passen sich flexibel an die Bildschirmgröße an und bleiben gut bedienbar.

Die Startseite der KIMaster-Plattform begrüßt den Benutzer und bietet eine Übersicht der verfügbaren Optionen und Spiele.

### 3.2.1 Navigationsleiste

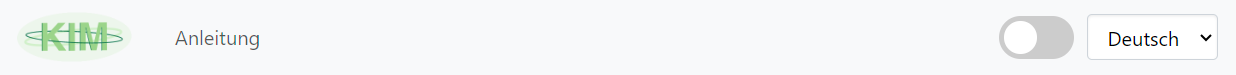


Abbildung 5: Navigationsleiste

In der linken oberen Ecke befindet sich das KIM-Logo, das auch als Button zur Rückkehr auf die Startseite dient.

Neben dem Logo ist der Link zur [Anleitung](#_4_Benutzeranleitung_für). Dieser führt zu Benutzeranleitung für die Schnittstelle (API).

In der oberen rechten Ecke gibt es einen Sprachschalter, um die Sprache der Plattform zu ändern. Die verfügbaren Sprachen sind Deutsch, Englisch, Spanisch, Französisch.

Links neben der Sprachauswahl, befindet sich auch der Schalter für die verschiedenen Farbmodi, wie z.B. Darkmode. Diese sind zur Zeit noch nicht integriert.

### 3.2.2 Begrüßung

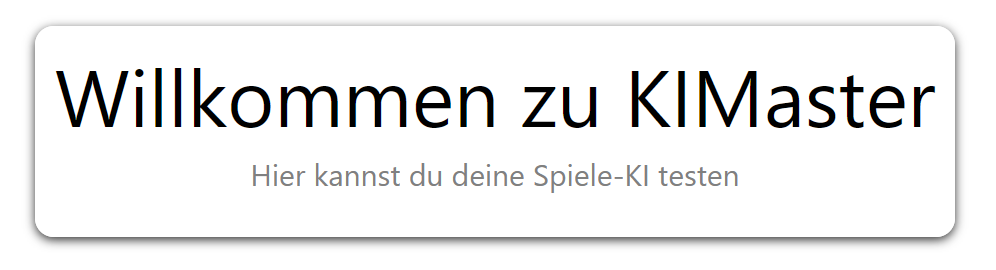


Abbildung 6: Begrüßung

Die Startseite zeichnet sich unter anderem dadurch aus, dass der Benutzer auf der Plattform willkommen geheißen wird. So hebt sich die Startseite auch von den anderen Seiten ab und man erkennt sofort, dass man sich auf dieser Seite befindet.

### 3.2.3 Spielauswahl

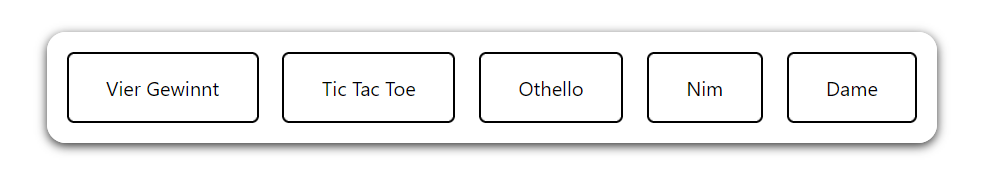


Abbildung 7: Spieleauswahl

Unterhalb der Begrüßung befindet sich die Spielauswahl. Hier sind die verfügbaren Spiele in anklickbaren Buttons nebeneinander aufgelistet. Jeder Button ist mit dem Namen des Spiels beschriftet. Die aktuell verfügbaren Spiele sind: Vier gewinnt, Tic Tac Toe, Othello, Nim und Dame.

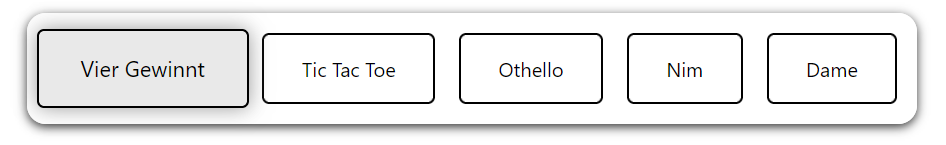


Abbildung 8: Spieleauswahl Hovering

Beim Hovern mit der Maus über die Buttons, wird der Button des Ausgewählten Spiels vergrößert und grau hervorgehoben. Hier im Beispiel am Vier Gewinnt Button dargestellt. Durch Klicken auf einen dieser Buttons gelangt der Benutzer in die jeweilige [Spielelobby](#_3.4_Lobby-Seite).

### 3.2.4 Eingabe des Lobbyschlüssels



Abbildung 9: Lobby beitreten

Unter der Spielauswahl gibt es ein Eingabefeld mit der Aufschrift „Lobby-Schlüssel eingeben“. Hier kann der Benutzer einen Schlüssel eingeben, um einer bestimmten Lobby beizutreten. Direkt daneben befindet sich ein weiterer Button mit der Aufschrift „Lobby beitreten“. Dieser Button ermöglicht es dem Benutzer, einer [bestehenden Lobby beizutreten](#_3.3_Warteraum), nachdem ein Schlüssel eingegeben wurde.

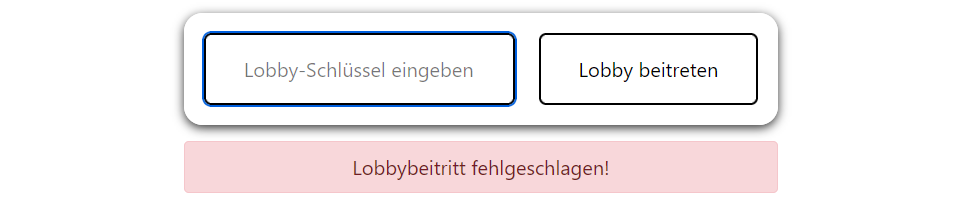


Abbildung 10: Lobby beitreten - Fehleingabe

Falls die Eingabe nicht korrekt war, erhält man eine Fehlermeldung unter dem Eingabefenster. Diese Fehlermeldung ist rot hervorgehoben und wird für 5 Sekunden eingeblendet, bevor sie wieder verschwindet.

### 3.2.5 Footer



Abbildung 11: Footer

Am unteren Rand der Seite befindet sich der Footer. Hier sind wichtige rechtliche Informationen verlinkt. Zum einen ist die [Webseite der THM](https://www.thm.de/site/) verlinkt, da dieses Projekt und die Plattform im Rahmen eines THM-Projekts entstanden sind. Außerdem finden Sie hier die Datenschutzbestimmungen und das Impressum.

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Abbildung 12: Datenschutzerklärung

Auf unserer Webseite legen wir großen Wert auf den Schutz Ihrer persönlichen Daten. Unsere Datenschutzrichtlinien erläutern, wie wir Ihre Daten erfassen, verwenden und schützen. Bitte lesen Sie unsere Datenschutzbestimmungen sorgfältig, um mehr darüber zu erfahren, wie wir Ihre Privatsphäre wahren.

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Abbildung 13: Impressum THM

Für rechtliche Informationen und Kontaktangaben verweisen wir auf das [Impressum der THM](https://www.thm.de/site/impressum.html) verwiesen. Dort finden Sie alle relevanten Details gemäß den gesetzlichen Anforderungen.

## 3.3 Warteraum

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Abbildung 14: Warteraum: belegte Position Spieler 1

Nach Eingabe des Lobby-Schlüssels auf der Hauptseite gelangen Sie in den Warteraum. Hier haben Sie die Möglichkeit, ihre Position im späteren Spiel zu konfigurieren. Je nach noch unbelegten Modi ist es möglich als Spieler oder Zuschauer beizutreten. So können sie als Spieler aktiv am Spiel teilnehmen. Als Zuschauer kann das Spielgeschehen beobachtet werden, sodass neben dem Beiwohnen zu normalen Partien zwischen zwei Nutzern auch Runden gegen KI-Gegner oder unsere KI-Implementierung KIM beobachtet werden können. Sie bekommen aktiv Feedback, wenn die ausgewählte Position bereits belegt ist. Die Anzahl der Zuschauer ist dabei nicht begrenzt.

Sobald alle Teilnehmer bereit sind und das Spiel gestartet wird, gelangen Sie automatisch zur [Spieleseite](#_3.5_Spielseite), wo das eigentliche Spiel stattfindet.

## 3.4 Lobby-Seite

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

Abbildung 15: Lobby-Seite

Die Lobby-Seite dient als zentrale Anlaufstelle für Ihre Spielekonfiguration. Die Spielelobbys sind allgemein gleich aufgebaut, mit leichten Anpassungen bezüglich der Schriftzüge und Spielekonfigurationsmöglichkeiten.

### 3.4.1 Spielekonfiguration

Bei der Spielekonfiguration können Sie die Parameter für Ihr Spiel festlegen. Passen Sie die Einstellungen nach Ihren Wünschen an.

Ein Bild, das Text, Screenshot, Schrift, Reihe enthält.

Automatisch generierte Beschreibung

Abbildung 16: Spielerkonfiguration

Ein Bild, das Text, Screenshot, Schrift, Reihe enthält.

Automatisch generierte Beschreibung

Abbildung 17: Spielmoduskonfiguration

Es besteht die Möglichkeit die eigenen Spielposition auf Spieler 1, Spieler 2 oder Zuschauer zu konfigurieren. Zudem kann der Spielmodus ausgewählt werden. So bieten wir normale „Spieler gegen Spieler“ Erlebnisse, aber auch „Spieler gegen KIM“ (unsere vortrainierte KI) oder auch das antreten mit der eigenen Spieler-KI an. Natürlich gibt es auch hier nochmal alle möglichen Konfigurationen.

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

Abbildung 18: Schwierigkeitskonfiguration

Wird ein Spiel gegen KIM ausgewählt, so kann noch der Schwierigkeitsgrad von KIM zwischen einfach, mittel und schwer verstellt werden.

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

Abbildung 19: Spielelobbywechsel

Möchte man zudem vielleicht doch das Spiel, als das zuvor ausgewählte, wechseln, so ist dies über einen eigenes Drop-Down-Menü einfach möglich. Die Spiele-Lobby muss dafür nicht verlassen werden und bereits weitergegebene Schlüssel können beibehalten werden.

### 3.4.2 Status und Lobby-Schlüssel

Ein Bild, das Text, Screenshot, Schrift, weiß enthält.

Automatisch generierte Beschreibung

Abbildung 20: Lobby-Schlüssel & QR-Code

Der Lobbyschlüssel ist ein eindeutiger Code, der Ihnen den Zugang zu einer bestimmten Lobby ermöglicht. Geben Sie diesen Schlüssel an Freunde und mitspielende oder Ihre eigene KI-Implementierung weiter, um der gewünschten Lobby beizutreten. Achten Sie darauf, den Schlüssel korrekt einzugeben, um Probleme beim Zugang zur Lobby zu vermeiden.

Ein Bild, das Schrift, Screenshot, Quadrat, weiß enthält.

Automatisch generierte Beschreibung

Abbildung 21: QR-Code

Der Einfachheit halber gibt es zusätzlich einen QR-Code, der das Scannen mit mobilen Geräten ermöglicht und das Beitreten in die Lobby vereinfacht.

Ein Bild, das Text, Screenshot, Schrift, weiß enthält.

Automatisch generierte Beschreibung

Abbildung 22: Positionsanzeige

Zusätzlich können Sie hier Ihren Status und den Status anderer Spieler einsehen, um sich einen Überblick darüber zu verschaffen, welche Positionen noch unbelegt sind und wie viele Zuschauer Sie gegebenenfalls bei Ihrem Spiel erwarten können.

Ein Bild, das Text, Screenshot, Schrift, Rechteck enthält.

Automatisch generierte Beschreibung

Abbildung 23: Lobby-Aktionen

Mit einem Klick auf Spiel Starten wird man auf die [Spieleseite](#_3.5_Spielseite) weitergeleitet. Alle weiteren Nutzer, die sich im [Warteraum](#_3.3_Warteraum) befunden haben, werden nun ebenfalls auf das Spiel weitergeleitet.

## 3.5 Spielseite

Ein Bild, das Screenshot, Text, Zahl, Taschenrechner enthält.

Automatisch generierte Beschreibung

Abbildung 24: Spieleseite)

Auf der Spieleseite finden Sie das zentrale Spielfeld, das sich in der Mitte der Seite befindet. Diese Ansicht zeigt nur die relevanten Schaltflächen und Funktionen an, die für das aktuelle Spiel notwendig sind. Zudem ist über dem Spielbrett angezeigt, wer gerade am Zug ist. Die Benutzeroberfläche ist so gestaltet, dass sie Ihnen eine klare und fokussierte Spielerfahrung bietet.

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

Abbildung 25: Spielregeln

Ein Button ermöglicht es Ihnen, die Spielregeln jederzeit einzusehen. Dieser Button hilft Ihnen, sich über die Spielmechanik und spezifische Regeln zu informieren, um strategische Entscheidungen zu treffen.

Mit dem Aufgeben-Button können Sie eine laufende Partie beenden, wenn Sie sich entscheiden, das Spiel aufzugeben. Dies beendet das aktuelle Spiel und zeigt Ihnen das [Ergebnis](#_3.5.4_Fertiges_Spiel) an.

|  |  |
| --- | --- |
| Ein Bild, das Screenshot, Muster, Farbigkeit, Rechteck enthält.  Automatisch generierte Beschreibung | Ein Bild, das Screenshot, Muster, Kreis, Design enthält.  Automatisch generierte Beschreibung |

Abbildung 26: Zurückgenommener Zug

Dieser Button ist nur in der Konfiguration „Spieler gegen KIM“ verfügbar. Er erlaubt es Ihnen, den letzten Zug rückgängig zu machen, um Ihre Strategie anzupassen oder Fehler zu korrigieren. Dabei wird automatisch natürlich auch der letzte Zug von KIM rückgängig gemacht.

### 3.5.1 Spielbrett

Das Spielbrett ist die zentrale Fläche, auf der alle Aktionen stattfinden. Hier sind die wichtigsten Interaktionen erklärt:

Ein Bild, das Screenshot, Muster, Kreis, Farbigkeit enthält.

Automatisch generierte Beschreibung

Abbildung 27: Valide Züge

* **Rechtsklick:** Wenn Sie mit der rechten Maustaste auf das Spielbrett klicken, wird eine Übersicht aller möglichen Züge angezeigt. Diese Funktion hilft Ihnen dabei, Ihre Optionen zu überprüfen und strategische Entscheidungen zu treffen.

Ein Bild, das Muster, Screenshot, Design, Rechteck enthält.

Automatisch generierte Beschreibung

Abbildung 28: Maus-Hoover

* **Maus-Hover:** Wenn Sie die Maus über ein Feld bewegen, wird dieses farbig hervorgehoben. Diese visuelle Markierung zeigt Ihnen an, welche Felder für Ihren nächsten Zug verfügbar sind und erleichtert die Orientierung auf dem Brett.

|  |  |
| --- | --- |
|  |  |

Abbildung 29: Ausgeführter Zug

* **Klick zum Ausführen:** Durch einen einfachen Klick auf ein hervorgehobenes Feld führen Sie den gewünschten Zug aus. Dieser Klick bestätigt Ihre Wahl und setzt die Spielaktion in Gang. Werden Spiele gegen KIM ausgeführt, setzt die KI daraufhin ihren Zug. Andernfalls ist der gegnerische Spieler oder dessen KI am Zug.

Diese Interaktionsmöglichkeiten sorgen für eine intuitive und benutzerfreundliche Steuerung des Spiels.

#### Besonderheiten

Beim Spiel „Dame“ gibt es spezifische Interaktionsschritte, die Sie beachten sollten:

Ein Bild, das Kreis, Quadrat, Brettspiel, Screenshot enthält.

Automatisch generierte Beschreibung

Abbildung 30: Dame-Brett

* **Spielstein auswählen:** Bevor Sie einen Zug ausführen können, müssen Sie zuerst den Spielstein anklicken, den Sie bewegen möchten. Dies aktiviert die Auswahl dieses Steins und zeigt Ihnen die möglichen Züge an.

Ein Bild, das Kreis, Quadrat, Screenshot, Brettspiel enthält.

Automatisch generierte Beschreibung

Abbildung 31: Mögliche Züge anzeigen

* **Mögliche Züge anzeigen:** Nachdem Sie einen Spielstein ausgewählt haben, werden alle gültigen Züge für diesen Stein auf dem Brett hervorgehoben. Diese Züge werden durch farbige Markierungen angezeigt, die Ihnen eine klare Übersicht über die verfügbaren Optionen bieten.

Ein Bild, das Kreis, Quadrat, Brettspiel, Hallensportarten enthält.

Automatisch generierte Beschreibung

Abbildung 32: Zug ausgeführt (Dame)

* **Zug ausführen:** Wählen Sie eines der markierten Felder aus, um Ihren Zug auszuführen. Ein Klick auf das gewünschte Feld bewegt den ausgewählten Spielstein dorthin und setzt das Spiel fort.

Zudem gibt es bei „Nim“ eine spezielle Vorgehensweise für das Ausführen von Zügen:

Ein Bild, das Text, Screenshot, Zahl, Uhr enthält.

Automatisch generierte Beschreibung

Abbildung 33: Nim-Brett

Ein Bild, das Text, Screenshot, Diagramm, Reihe enthält.

Automatisch generierte Beschreibung

Abbildung 34: Ausgewählte Reihe

* **Reihe und Menge auswählen:** Klicken Sie auf die Reihe, aus der Sie Streichhölzer bzw. Stäbchen entfernen möchten. Sobald Sie eine Reihe ausgewählt haben, wird oben auf einem Button angezeigt, wie viele Streichhölzer oder Stäbchen Sie in dieser Reihe wegnehmen möchten. Durch mehrfaches Klicken auf die Reihe wird die Anzahl der Hölzer jedes Mal um eins erhöht.

Ein Bild, das Text, Screenshot, Diagramm, Reihe enthält.

Automatisch generierte Beschreibung

Abbildung 35: Zug bestätigt

* **Zug bestätigen:** Erst wenn Sie auf den Button mit der Anzahl der ausgewählten Streichhölzer oder Stäbchen klicken, wird der Zug ausgeführt. Dieser Klick bestätigt Ihre Wahl und entfernt die angegebenen Streichhölzer bzw. Stäbchen aus der gewählten Reihe.

### 3.5.4 Fertiges Spiel

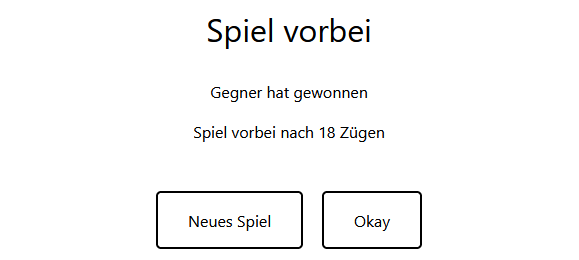


Abbildung 36: Spiel vorbei

Wenn das Spiel endet – sei es durch Aufgabe eines Spielers, das Fehlen weiterer gültiger Züge (wie bei Othello) oder den Sieg eines Spielers – wird ein Pop-up angezeigt. Dieses Pop-up informiert Sie darüber, wer das Spiel gewonnen hat, und gibt die Anzahl der gespielten Züge an. Sie haben dann die Möglichkeit, entweder ein neues Spiel zu starten oder durch Klicken auf „Okay“ zu den weiteren Spielfunktionen zurückzukehren. Das Pop-up ermöglicht es Ihnen, schnell zu einer neuen Runde zu wechseln oder das Spielgeschehen zu beenden und andere Optionen zu erkunden.

Ein Bild, das Screenshot, Text, Diagramm, Rechteck enthält.

Automatisch generierte Beschreibung

Abbildung 37: Spielende zusätzliche Funktionen

Hat man sich dazu entschieden nicht direkt eine neue Runde durch „Neues Spiel“ zu starten, so landet man wieder auf der Spielseite mit dem letzten Stand des Spielbrettes. Es sind dabei einige neue Buttons aufgetaucht, die neue Funktionen bereitstellen.

Neben „Spiel verlassen“, „Neues Spiel“ und „Lobby“ gibt es noch zwei Funktionalitäten, die genauer erläutert werden.

### 3.5.5 Zeitleiste

|  |  |  |
| --- | --- | --- |
|  |  |  |

Abbildung 38: Zeitleiste - vorherig

Am unteren Rand des Spielfeldes ist die Zeitleisten-Funktionalität eingeblendet worden. Durch Betätigen der Buttons navigiert man durch die einzelnen Züge des vergangenen Spiels und kann sich so jeden Zug und jede Aktion noch einmal in Ruhe anschauen.

### 3.5.6 Fehleranalyse

Ein Bild, das Text, Screenshot, Rechteck, Quadrat enthält.

Automatisch generierte Beschreibung

Abbildung 39: Fehleranalyse

Eine weitere wichtige Funktionalität stellt die Fehleranalyse dar. Betätigen Sie den Button am oberen Rand, so öffnet sich ein Pop-Up mit einer Auswertung von schlecht gewählten Zügen. Diese sind ebenfalls als Button realisiert, sodass ein Klick darauf das zugehörige Spielbrett mit dem Spielstand zum Zeitpunkt des schlechten Zuges anzeigt. Um sich die Situation noch einmal genauer anzuschauen kann hier auch wieder durch das Spiel iteriert werden mit „Nächstes“ und „Vorherig“.

### 3.5.7 Mini-Fenster

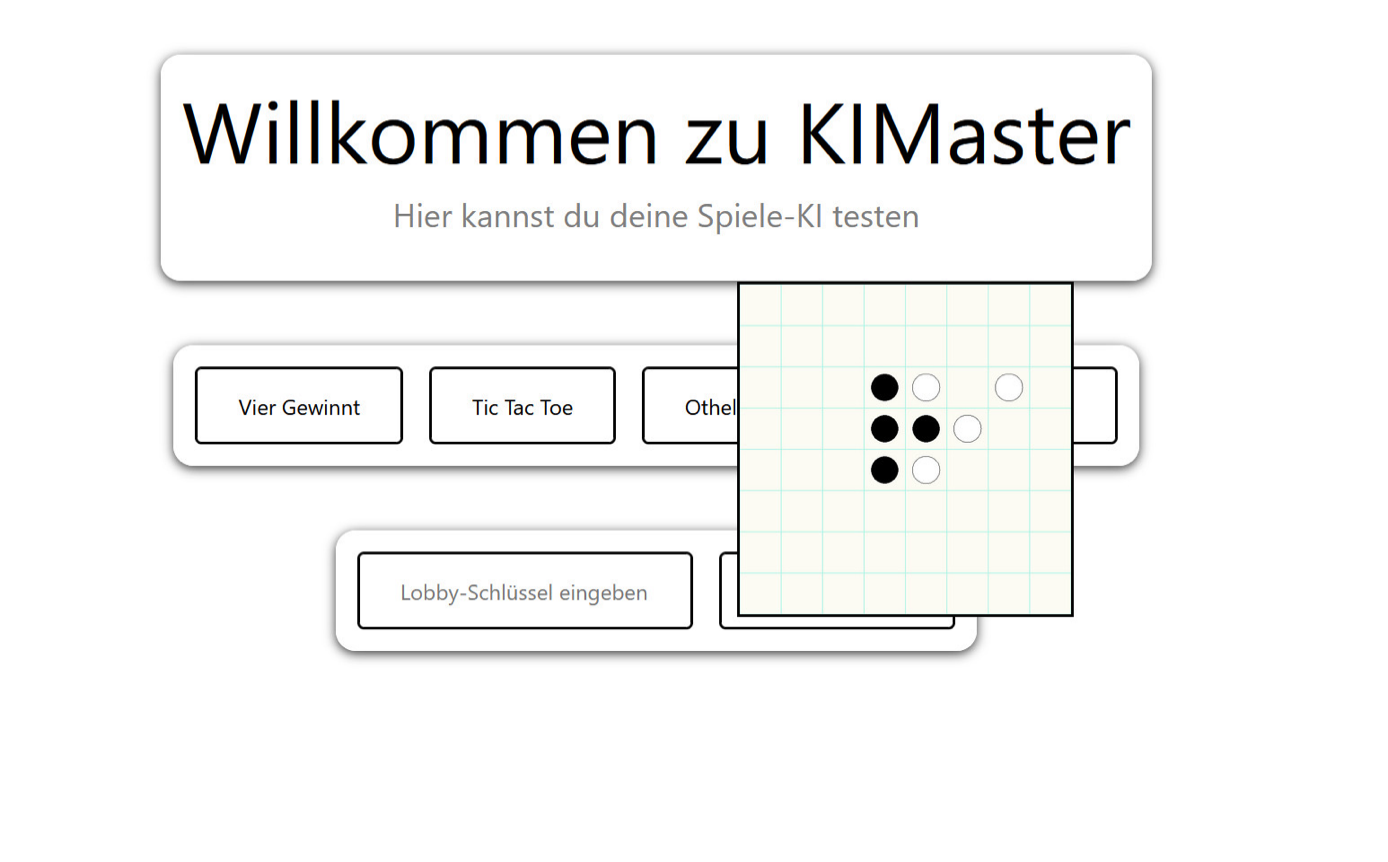


Abbildung 40: Mini-Fenster

Eine weitere spaßige Funktion stellt das Mini-Fenster da. Der Spieler hat die Möglichkeit aus der Spielseite rauszugehen, ohne dass das Spiel beendet werden muss. Man kann so durch die Plattform navigieren und jederzeit das Spiel fortsetzen.

Dieses Fenster taucht dabei automatisch auf, sobald man beispielsweise auf den Logo-Button der Webseite (oben links) klickt. Es lässt sich verschieben und ebenfalls zum Spiel zurückkehren durch einen Doppelklick auf das Fenster.

Diese Funktion ist ausschließlich plattformintern nutzbar und steht nicht mehr zur Verfügung, sobald man diese verlässt.

# 4 Benutzeranleitung für die KIMaster-Schnittstelle

Die KIMaster-Schnittstelle ist ein Framework zur Verwaltung der Kommunikation zwischen einem Client und einem WebSocket-Server. Sie bietet Methoden zur Verbindung, zum Senden und Empfangen von Nachrichten sowie zur Verwaltung der Verbindungslebensdauer. Diese Schnittstelle wurde speziell für die Nutzung mit dem [FastAPI](https://fastapi.tiangolo.com/) Server des Projekts KIMaster entwickelt.

## 4.1 Überblick über die Schnittstelle (Python Beispiel)

Die KIMaster-Schnittstelle ermöglicht es, einfach und effizient mit einem WebSocket-Server zu kommunizieren. Es können verschiedene Befehle gesendet und Antworten erhalten werden in Echtzeit. Dabei ist zu beachten, dass ein Befehl auch mehrere in nicht geordnete Antworten mit sich bringen kann. Hier ist eine **Asynchronität** zu beachten.

**Hauptfunktionen:**

* Verbindung zu einem WebSocket-Server herstellen.
* Befehle an den Server senden.
* Nachrichten vom Server empfangen.

Es werden Klassen und Code-Abschnitte zur einfacheren Anbindung bereitgestellt. Dabei enthält dieKIMaster-Klasse die grundlegende Infrastruktur für die WebSocket-Kommunikation, einschließlich Verbindungsmanagement, Befehlssendung und Nachrichtenerhalt. Die Example-Klasse erweitert diese Funktionalität, indem sie spezifische Befehle und Szenarien implementiert, die für die Anwendung erforderlich sind.

**KIMaster-Klasse:**

* **\_\_init**\_\_: Initialisiert die Klasse mit einer Liste von URIs.
* **connect**: Stellt eine Verbindung zu einem WebSocket-Server her.
* **send\_cmd**: Sendet Befehle an den Server.
* **receive**: Empfängt Nachrichten vom Server.
* **close**: Schließt die Verbindung.
* **run**: Führt eine Coroutine aus.
* **handler**: Verwaltet das Senden und Empfangen von Nachrichten.
* **print\_message**: Formatiert und druckt Nachrichten.
* **show**: Zeigt ein Bild aus einem Byte-Stream an.

**Example-Klasse:**

* **\_\_init\_\_:** Initialisierung der Klasse.
* **input\_thread**: Sammeln und Verarbeiten von Benutzereingaben.
* **send\_handler:** Asynchrones Senden von Befehlen an den Server.
* **receive\_handler:** Asynchrones Empfangen und Verarbeiten von Nachrichten vom

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Abbildung 41: Ordnerstruktur Python Schnittstelle

## 4.2 Kommandos

Die Schnittstelle bietet verschiedene Befehle, die an den WebSocket-Server gesendet werden können. Diese Befehle werden als [JSON](https://www.json.org/json-de.html)-Nachrichten formatiert und enthalten spezifische Schlüssel und Daten, die vom Server verarbeitet werden. Alle Befehle sind in vollem Umfang mit ihren Möglichkeiten, Variationen und optionalen Parametern in der [commands.md](https://github.com/12ghostrider21/KIMaster/blob/main/Spezifikation/Dokumentation/Sonstiges/commands.md) enthalten. Zu finden unter: Spezifikation/Dokumentation/Sonstiges/.

* Struktur einer zu sendenden Nachricht im JSON-Format:

{“command“: command,

“command\_key“: commandy\_key,

“data“: data}

* Struktur einer zu empfangenden Nachricht im JSON-Format:

{“response\_code“: response\_code,

“response\_msg“: response\_msg,

“data“: data}

* Beispiel für Kommandos:
  + Lobby: Statusabfrage, Positionierung, Erstellung und Verlassen von Lobbys.
  + Play: Erstellen eines Spiels mit bestimmten Parametern (z.B. Spielname, Schwierigkeitsgrad, Modus)

## 4.3 Voraussetzungen

Bevor begonnen werden kann, muss sichergestellt werden, dass die folgenden Voraussetzungen erfüllt sind:

**Voraussetzungen**

* [Python 3.7](https://www.python.org/downloads/) oder höher (erstellt wurde mit 3.11)
* Abhängigkeiten: asyncio, json, threading, queue, PIL, websockets

## 4.4 Einrichtung

1. **Installieren der notwendigen Bibliotheken**: Um die KIMaster-Schnittstelle zu verwenden, müssen die erforderlichen Python-Bibliotheken installiert sein. Ist dies nicht der Fall muss folgendes durchgeführt werden: Das Terminal öffnen und ausführen des folgenden Befehls zur Installation der notwendigen Bibliotheken:

Ein Bild, das Text, Schrift, Screenshot, Grafiken enthält.

Automatisch generierte Beschreibung

Abbildung 42: Install-Befehl Bibliotheken

1. **Verbindung zum WebSocket-Server**: Verwenden Sie die KIMaster-Klasse, um eine Verbindung zum Server herzustellen. Dies wird durch die Methodeconnect() der KIMaster-Klasse hergestellt. Diese Methode versucht, eine Verbindung zu einer der URIs herzustellen. Die erste erfolgreiche Verbindung wird genutzt und nicht weiter getestet. Mit der Liste können mehrere Verbindungen hinterlegt werden. Beispielsweise die localhost Adresse zum Testen und die offizielle Adresse auf der Deployt wird.

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Abbildung 43: Zu verbindende URIs

Hier gut zu erkennen, die erste URI ist die der offiziellen KIMaster Adresse. Die Zweite ist die localhost Adresse, wenn das System bei einem Lokal in Docker gestartet wird.

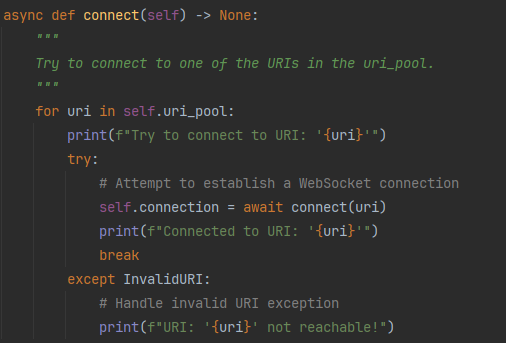


Abbildung 44: connect()

1. **Einbindung der Schnittstelle:** Mit der handler-Methde, können zwei weitere Methoden verlinkt werden. Diese werden dann von der KIMaster-Klasse asynchron parallel ausgeführt.

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Abbildung 45: main()

## 4.5 Nutzung der Hauptfunktionen

DieKIMaster-Schnittstelle bietet mehrere Hauptfunktionen, die für die Kommunikation mit dem WebSocket-Server verwendet werden.

* **Befehle senden:** Die Methode send\_cmd() wird verwendet, um Befehle an den WebSocket-Server zu senden. Sie nimmt den Befehl, den Befehlsschlüssel und optional zusätzliche Daten als Parameter entgegen.

Ein Bild, das Text, Screenshot, Software enthält.

Automatisch generierte Beschreibung

Abbildung 46: send\_cmd()

* **Nachrichten empfangen**: Die Methodereceive() wird verwendet, um Nachrichten vom WebSocket-Server zu empfangen. Sie kann Nachrichten im JSON-Format, als Zeichenkette oder als Byte-Stream zurückgeben.

Ein Bild, das Text, Screenshot enthält.

Automatisch generierte Beschreibung

Abbildung 47: receive()

* **Hauptprogramm**: Das Hauptprogramm initialisiert die Example-Klasse, startet die WebSocket-Verbindung und verwaltet die asynchronen Aufgaben für das Senden und Empfangen von Nachrichten. Dazu muss diese ausgeführt werden.

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Abbildung 48: Hauptprogramm

# 5 Häufige Probleme und Lösungen

## 5.1 Meine Verbindung bricht ständig ab, wenn ich etwas empfange?

Da beim Empfangen nicht vorher bekannt ist was das Format der empfangenen Daten ist, sollte vorerst überprüft werden, ob dieses richtig erkannt wurden.



Abbildung 49: Formatüberprüfung

Bei bytes handelt es sich um ein PNG Bild, das interpretiert werden muss, bei str ein normaler String und bei dict um ein JSON-Objekt, von dem der Antwortcode, die -nachricht und der mögliche zusätzliche Anhang genutzt werden kann.

## 5.2 Ich kann mich nicht mit dem THM-Server verbinden?

Befinden Sie sich im THM-Netzwerk und können Sie sich auch per Webseite https://kimaster.mni.thm.de verbinden? Die URI von extern lautet: wss./kimaster.mni.thm.de/ws

## 5.3 Ich erhalte keine Antworten trotz While True loop?

Da asynchrone Parallelität trotz handler-Methode nicht immer funktionieren, probieren Sie in den verlinkten Sende- oder Empfangs-Methoden ein asyncio.sleep(0.1)einzubauen, um dem System die Möglichkeit zu geben beide Routinen zu bearbeiten.

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Abbildung 50: receive\_handler() ohne sleep

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Abbildung 51: send\_handler() mit sleep

# 6 Literaturverzeichnis

1. Apple. *Safari*. [Online]. Verfügbar unter: <https://www.apple.com/de/safari/> [Zugriff am: 24. Juli 2024].
2. Microsoft. *Microsoft Edge*. [Online]. Verfügbar unter: <https://www.microsoft.com/de-de/edge?form=MA13FJ> [Zugriff am: 24. Juli 2024].
3. THM. *THM - Technische Hochschule Mittelhessen*. [Online]. Verfügbar unter: <https://www.thm.de/site/> [Zugriff am: 24. Juli 2024].
4. THM. *THM Campusnetz VPN - Cisco AnyConnect*. [Online]. Verfügbar unter: <https://www.thm.de/its/campusnetz/vpn/ciscoanyconnect.html> [Zugriff am: 24. Juli 2024].
5. Google. *Google Chrome*. [Online]. Verfügbar unter: <https://www.google.com/intl/de_de/chrome/> [Zugriff am: 24. Juli 2024].
6. Mozilla. *Firefox*. [Online]. Verfügbar unter: <https://www.mozilla.org/de/firefox/> [Zugriff am: 24. Juli 2024].
7. Mozilla Developer Network. *WebSockets API - MDN Web Docs*. [Online]. Verfügbar unter: <https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API> [Zugriff am: 24. Juli 2024].
8. Oracle. *Java*. [Online]. Verfügbar unter: <https://www.java.com/de/> [Zugriff am: 24. Juli 2024].
9. Microsoft. *C#-Lehrgang - Microsoft Learn*. [Online]. Verfügbar unter: <https://learn.microsoft.com/de-de/dotnet/csharp/tour-of-csharp/> [Zugriff am: 24. Juli 2024].
10. Rust Project Developers. *Rust*. [Online]. Verfügbar unter: <https://www.rust-lang.org/> [Zugriff am: 24. Juli 2024].
11. Kotlin. *Kotlin*. [Online]. Verfügbar unter: <https://kotlinlang.org/> [Zugriff am: 24. Juli 2024].
12. Python Software Foundation. *Python*. [Online]. Verfügbar unter: <https://www.python.org/> [Zugriff am: 24. Juli 2024].
13. THM. *Impressum der THM*. [Online]. Verfügbar unter: <https://www.thm.de/site/impressum.html> [Zugriff am: 24. Juli 2024].
14. Tiangolo. *FastAPI*. [Online]. Verfügbar unter: <https://fastapi.tiangolo.com/> [Zugriff am: 24. Juli 2024].
15. JSON. *JSON - JSON*. [Online]. Verfügbar unter: <https://www.json.org/json-de.html> [Zugriff am: 24. Juli 2024].
16. Python Software Foundation. *Python Downloads*. [Online]. Verfügbar unter: <https://www.python.org/downloads/> [Zugriff am: 24. Juli 2024].
17. 12ghostrider21. *KIMaster - Dokumentation*. [Online]. Verfügbar unter: <https://github.com/12ghostrider21/KIMaster/blob/main/Spezifikation/Dokumentation/Sonstiges/commands.md> [Zugriff am: 24. Juli 2024].