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 |
| 0.3 | 21.07.2024 | 3 | Schreiben | Omar Karkotli |

|  |  |  |
| --- | --- | --- |
| Herausgeber | [Technische Hochschule Mittelhessen – FB06 Mathematik, Naturwissenschaften und Informatik](https://www.thm.de/mni/) | |
| Dateiname | Benutzerhandbuch\_ver\_0.4 | |
| Dokumentenbezeichnung | Technische Dokumentation: Plattform zum Vergleich von Spiele-KIs | |
| Version | 0.4 | |
| Stand | Dienstag, 23. Juli 2024 | |
| Status | In Bearbeitung | |
| 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. Technische Dokumentation.“ | |

Inhaltsverzeichnis

[1 Einführung 6](#_Toc172645297)

[1.1 Kurzübersicht der Software 6](#_Toc172645298)

[1.2 Zielgruppen 6](#_Toc172645299)

[2 Einrichtung 7](#_Toc172645300)

[2.1 Systemanforderungen 7](#_Toc172645301)

[3 Benutzeranleitung für die Webseite 8](#_Toc172645302)

[3.1 Webseite aufrufen 8](#_Toc172645303)

[3.2 Startseite 9](#_Toc172645304)

[3.2.1 Navigationsleiste 9](#_Toc172645305)

[3.2.2 Begrüßung 10](#_Toc172645306)

[4 Benutzeranleitung für die KIMaster-Schnittstelle 10](#_Toc172645307)

[4.1 Überblick über die Schnittstelle (Python Beispiel) 10](#_Toc172645308)

[4.2 Kommandos 11](#_Toc172645309)

[4.3 Voraussetzungen 12](#_Toc172645310)

[4.4 Einrichtung 12](#_Toc172645311)

[4.4 Nutzung der Hauptfunktionen 14](#_Toc172645312)

[5 Häufige Probleme und Lösungen 15](#_Toc172645313)

[5.1 Meine Verbindung bricht ständig ab, wenn ich etwas empfange? 15](#_Toc172645314)

[5.2 Ich kann mich nicht mit dem THM Server verbinden? 15](#_Toc172645315)

[5.3 Ich erhalte keine Antworten trotz While True loop: 15](#_Toc172645316)

Abbildungssverzeichnis

**Es konnten keine Einträge für ein Abbildungsverzeichnis gefunden werden.**

Tabellenverzeichnis

**Es konnten keine Einträge für ein Abbildungsverzeichnis gefunden werden.**

# 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 Chrome, Firefox, Safari oder Edge ist erforderlich.
* **Netzwerkverbindung**: Der Zugriff muss innerhalb des Netzwerks der THM erfolgen. Dies kann entweder direkt vor Ort oder durch Nutzung eines VPN-Dienstes der THM 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 THM erfolgen, was entweder direkt vor Ort oder durch die Nutzung eines VPN-Dienstes der THM möglich ist. <https://www.thm.de/its/campusnetz/vpn/ciscoanyconnect.html>
* **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-Verbindungen realisiert. Stellen Sie sicher, dass Ihre Programmierumgebung und die verwendete Sprache die Einrichtung und Nutzung von WebSocket-Verbindungen unterstützen.

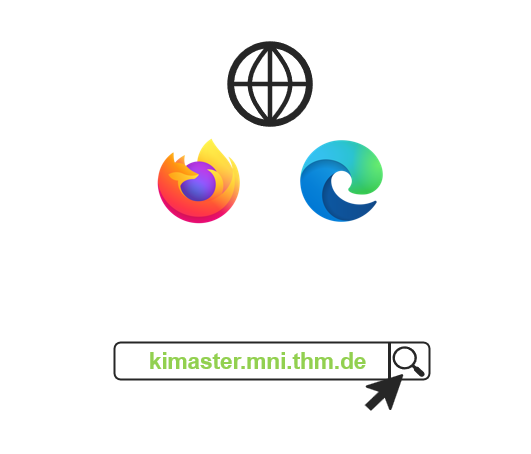
**Zusätzliche Hinweise**

* 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 oder kontaktieren Sie den IT-Support. <https://www.thm.de/its/campusnetz/vpn/ciscoanyconnect.html>
* Überprüfen Sie, ob zusätzliche Bibliotheken oder Frameworks für die Unterstützung von WebSocket-Verbindungen in Ihrer Programmierumgebung erforderlich sind, und installieren Sie diese entsprechend.
* Wir bieten eine Verbindungsbeispiel mit Python zur Verfügung.

[GitHub Repo: Extern / Python](https://github.com/12ghostrider21/Plattform-fuer-Vergleich-von-Spiele-KIs/tree/main/External/Python)

# 3 Benutzeranleitung für die Webseite

## 3.1 Webseite aufrufen

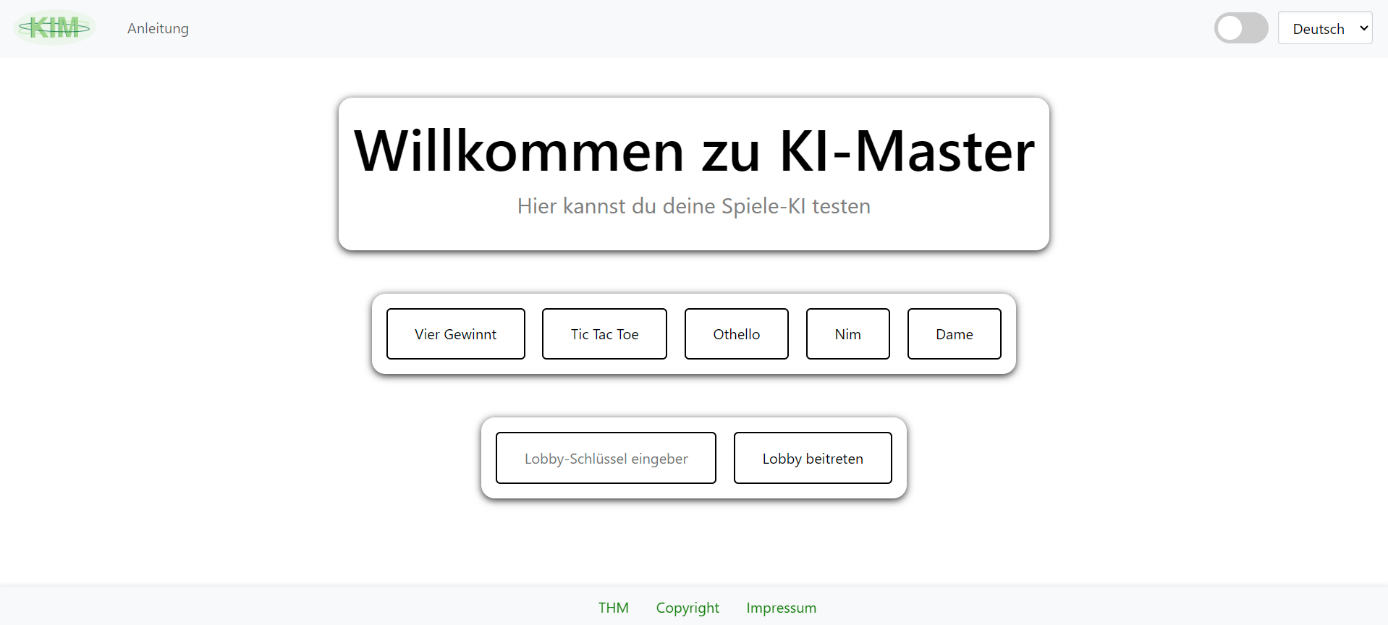


**Schritt 1:** Öffnen Sie einen Webbrowser Ihrer Wahl.

**Schritt 2:** Geben Sie die URL kimaster.mni.thm.de in die Adressleiste Ihres Browsers ein.

**Schritt 3:** Bestätigen Sie Ihre Eingabe durch Drücken der Eingabetaste.

## 3.2 Startseite



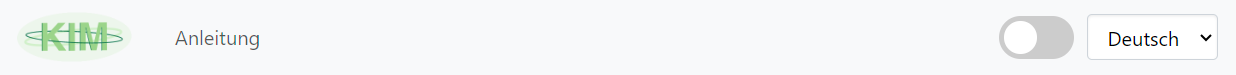
Die Startseite von KI-Master, kurz KIM genannt, ist übersichtlich und benutzerfreundlich gestaltet. Das Design ist minimalistisch und fokussiert sich darauf, dem Benutzer eine einfache Navigation und eine klare Struktur zu bieten.

(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 KI-Master-Plattform begrüßt den Benutzer und bietet eine Übersicht der verfügbaren Optionen und Spiele.

### 3.2.1 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.)

### 3.2.2 Begrüßung

# 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 FastAPIServer 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

## 4.2 Kommandos

Die Schnittstelle bietet verschiedene Befehle, die an den WebSocket-Server gesendet werden können. Diese Befehle werden als JSON-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/Plattform-fuer-Vergleich-von-Spiele-KIs/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 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.

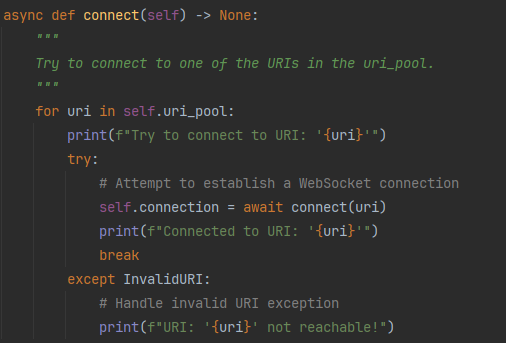
Automatisch generierte Beschreibung

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

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



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

## 4.4 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

* **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

* **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 dieses ausgeführt werden.

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

# 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 diese richtig erkannt wurden.



Bei bytes -> PNG Bild, beit str -> normaler String, bei dict -> json.

## 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 hander-Methode nicht immer funktionieren, probieren Sie in den Verlinkten sende oder Empfangs Methoden eine 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 BeschreibungEin Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung