Abschlussprojekt Softwaredesign

Von:
Philipp Höpperger, Niclas Jurisch

Einleitung

Das Abschlussprojekt im Bereich Softwaredesign hat das Ziel, **ebene Mechanismen zu berechnen und darzustellen**. Dazu wurde das Projekt in verschiedene **Teilmodule** unterteilt, um die Übersichtlichkeit und Bearbeitbarkeit zu optimieren. Die wesentlichen Bestandteile sind:

- database.py
- uistr_v1.py
- berechnungKopie.py
- user_db.py
- Die Datenbanken mechanisms.json und users.json

Unsere Lösung bietet eine **vollständige Simulationsebene für Mechanismen**, inklusive Berechnung, Speicherung und Darstellung.

Projektstruktur und Funktionsweise

1. Datenbankverwaltung (database.py)

In database.py wird das **Datenbank-Handling** mit **TinyDB** realisiert.

- Es wird überprüft, ob die erforderlichen **Speicherorte** für die Datenbanken existieren, andernfalls werden sie erstellt.
- Die Datenbank MechanismTable verwaltet Mechanismus-Daten mit folgenden Funktionen:
    - save_mechanism(): Speichern eines Mechanismus
    - load_mechanism(): Laden eines gespeicherten Mechanismus
    - delete_mechanism(): Löschen eines Mechanismus
    - list_mechanisms(): Auflisten aller gespeicherten Mechanismen

2. Benutzerverwaltung (user_db.py)

In user_db.py wird die **Benutzerkonten- und Passwortverwaltung** realisiert.

- Zur **sicheren Speicherung** von Passwörtern wird **hashlib** verwendet.
- Benutzer werden in der Datei users.json verwaltet.

3. Mechanismus-Speicherung (mechanisms.json und users.json)

- **users.json**: Speichert registrierte Benutzer mit gehashten Passwörtern.

- **mechanisms.json**: Speichert Mechanismen **benutzerindividuell**, sodass jeder Nutzer nur auf seine eigenen Mechanismen zugreifen kann.

Mechanismus-Berechnung (berechnungKopie.py)

Dieses Modul definiert die Kernklassen für die Mechanismus-Berechnung:

1. Kernklassen

- Joint: Repräsentiert ein Gelenk mit x/y-Koordinaten und einer Fixierungsoption.
- Link: Verbindet zwei Gelenke und speichert deren feste Verbindungslänge.
- Mechanism: Verwaltet eine Kombination aus Gelenken und Verbindungen sowie deren Berechnung.

2. Berechnungsfunktionen

- solve_positions(): Berechnet die Positionen beweglicher Elemente mit least_squares().
- rotate_crank(): Simuliert die Rotation einer Kurbel.
- update(): Aktualisiert die Bewegung für die Animation.
- animate(): Erstellt eine matplotlib-Animation des Mechanismus.

Web-Oberfläche (uistr_v1.py)

Dieses Skript implementiert eine interaktive Webanwendung mit Streamlit zur Mechanismus-Simulation.

1. Hauptfunktionen

- Benutzerverwaltung: Anmelden und Registrieren neuer Nutzer.
- Mechanismusverwaltung:
    - Speichern, Laden, Bearbeiten und Löschen von Mechanismen.
- Simulation und Animation:
    - Darstellung und Berechnung verschiedener Mechanismen.
    - Möglichkeit zur Nachverfolgung eines ausgewählten Gelenks.

Fazit

Das Projekt ermöglicht eine umfassende Erstellung, Berechnung und Visualisierung von ebenen Mechanismen. Durch die modulare Architektur ist es flexibel erweiterbar und bietet eine benutzerfreundliche Oberfläche zur Interaktion mit Mechanismusdaten. Die Kombination aus Datenbankverwaltung, Mechanismusberechnung und einer interaktiven Web-App stellt eine ganzheitliche Lösung für die Simulation von Mechanismen dar.

Anmerkung

Für Testzwecke und Einfachheit wurde bereits ein Nutzerkonto erstellt. Die Anmeldedaten sind:

Nutzername: test1

Passwort: 1

In diesem Nutzkonto ist zudem der Mechanismus „strandbeest“ gespeichert, der ein Beinmechanismus des Strandbeests auf Grundlage des gegebenen Wikipedia Artikels: https://de.wikipedia.org/wiki/Strandbeest simuliert.