

Skyjo Multiplayer Client

Architektur-Übersicht

Hauptkomponenten

`client.py` – Haupteinstiegspunkt

Zweck: Startet das Spiel und koordiniert alle Komponenten

- Threading:
 - o GUI im Hauptthread
 - o Netzwerkkommunikation im Hintergrundthread
- Zentrale Funktionen:
 - o `main_game_loop()`: Verarbeitet Spieleraktionen
 - o `connection_handler()`: Kommuniziert mit dem Server

`GUI.py` – Hauptcontroller

Zweck: Steuerung der gesamten Benutzeroberfläche

- Koordiniert: Alle anderen GUI-Komponenten
- Verwaltet: Menü-Zustände und Spielphasen

`GameStateManager.py` – Spiellogik

Zweck: Verwaltung des aktuellen Spielzustands

- Funktionen:
 - o Auswahl und Aktionen von Karten
 - o Verwaltung der Spielphasen (Initial vs. Normal)
 - o Spieleranweisungen

`GameRenderer.py` – Darstellung

Zweck: Visuelle Darstellung des Spiels

- Rendert:
 - o Spielerkarten & Stapel
 - o Spielinformationen
 - o Aktionsbuttons

EventHandler.py – Eingabeverarbeitung

Zweck: Verarbeitung von Benutzereingaben

- Behandelt:
 - Mausklicks auf Karte
 - Stapelinteraktionen
 - Hover-Effekte

WidgetManager.py – UI-Widgets

Zweck: Verwaltung der GUI-Widgets

- Menüs:
 - Hauptmenü
 - Spiel hosten
 - Spielbildschirm

network.py – Netzwerkkommunikation

Zweck: Austausch mit dem Spielserver

Funktionen:

- Verbindungsaufbau
- Senden und Empfangen von Daten

Kommunikation

Client → Server

Sendet Kommandos an den Server

- ("Check Draw Pile", True)
- ("Take from Draw Pile", (row, col))
- ("Take from Discard Pile", (row, col))
- ("Flip Card", (row, col))