

# GUI/Visualizer für Poker-Spiele

Motivationsschreiben im Modul Advanced Python, B TI, WiSe25/26

Dozent: Markus Schubert (markus.schubert@bht-berlin.de)

October 19, 2025

Friedemann Doll (Matr.-Nr.: 924315 s86446@bht-berlin.de)

## 1 Beschreibung

Ich arbeite aktuell an meiner Bachelorarbeit „Implementierung von Counterfactual Regret Minimization in zunehmend komplexen Pokervarianten“. Im Zentrum dieser Arbeit steht die Implementierung des CFR-Algorithmus in Python zur Berechnung von Nash-Gleichgewichts-Strategien für verschiedene Pokervarianten.

Hierfür habe ich mehrere Spielumgebungen gebaut, welche der CFR-Algorithmus dann nutzt, um eine Strategie zu finden und sie in einem pickle File zu speichern. Im pickle File ist ein Dictionary in welchem Aktionswahrscheinlichkeiten für jede mögliche Spielsituation dieser Pokervariante gespeichert sind. Ein Strategy Agent kann diese abgespeicherte Strategie nun in meiner Spielumgebung spielen. Dies tut er, indem er den aktuellen Spielzustand als Key für das Dictionary verwendet. Damit greift er auf Wahrscheinlichkeiten für die im aktuellen Zustand legalen Aktionen zu. Per Zufallssampling wird dann eine dieser Aktionen ausgewählt und ausgeführt.

In diesem Modul möchte als Extra Feature für den bestehenden Bachelorarbeit Code ein GUI/Visualizer entwickeln. Ich habe im Github meiner Bachelorarbeit

einen separaten Branch dafür angelegt, in welchem ich an der Hausarbeit in Latex und dem Code in Python arbeiten werde.

<https://github.com/Miedefran/CounterfactualRegretMinimizationPoker>

## **1.1 Idee**

Das GUI Projekt soll drei verschiedene Modi umfassen. Einen Visualizer für Agent gegen Agent Duelle mit Play, Pause, Step Forward und Step Backward Funktionen. Ein Modus in welchem ein Mensch lokal gegen einen Agenten spielen kann.

Und einen Multiplayer Modus in dem ein Mensch Online gegen einen anderen Mensch spielen kann.

## **1.2 Ansatz**

Die Architektur basiert auf einer Basis GUI Klasse, von der die drei verschiedenen Modi erben. Dadurch kann ich Teile des Codes wiederverwenden und die Benutzeroberfläche bleibt so konsistent wie möglich.

Die Agent vs Agent Visualisierung nutzt dann für die Step Mechaniken die vorhandenen Spielbaumtraversierungsmethoden der Game Klasse.

Die Agent vs Human Anwendung stellt zusätzlich eine Möglichkeit für den User Aktionen einzugeben, welche von der Game Logik verarbeitet werden.

Der Human vs Human Multiplayer soll einen Flask Server mit WebSockets verwenden, um es zwei Spielern zu ermöglichen in Echtzeit gegeneinander Poker zu spielen.

## **1.3 Technologien**

Als Technologien verwende ich PyQt6 für die Desktop GUI, Flask mit Flask-SocketIO für den WebSocket Server, und die bereits existierenden Spielumgebungen aus meiner Bachelorarbeit. Die Strategien werden aus den trainierten Pickle Dateien geladen und vom Strategy Agent verwendet.

## **1.4 Ausblick**

Mögliche Erweiterungen wären ein Scoreboard/Leaderboard für Human vs Agent und Human vs Human Spiele oder eine Strategievisualisierung, bei welcher dem User angezeigt wird, welche Strategie ein bestimmter Strategy Agent in dieser Situation spielen würde. Dies könnte man mittels einer einfachen Tabelle oder einer Heatmap tun.