

PyJack ♥♣♦♠

Analyse

Problem

Wir helfen Menschen, lange Wartezeiten und Langeweile zu überbrücken. Zusätzlich unterstützen wir Glücksspielende dabei, zu spielen, ohne das Risiko Geld zu verlieren und dennoch die Befriedigung des Spielens zu erleben.

Szenario

PyJack ist ein einfach zugängliches Spiel gegen Langeweile. Spielenden können jederzeit und überall ohne Internetverbindung spielen.

User stories:

- Als Nutzer möchte ich sofort nach Sieg oder Niederlage ein neues Spiel starten können.
- Als Nutzer möchte eine Karte ziehen oder bei der aktuellen Hand bleiben können.
- Als Nutzer möchte ich informiert werden, ob ich gewonnen oder verloren habe.
- Als Nutzer möchte ich, dass Gewinne und Verluste in einer Datei protokolliert werden, damit ich meine Gewinn-/Verlustrate verfolgen kann.

Use cases:

- Neues Spiel starten (Deck mischen, Karten austeilen)
- Spielzustand anzeigen (Hände mit Werten)
- Spielerentscheidung treffen (Hit/Stand mit Validierung)
- Dealer-Automatik (Standardregeln: Hit bei 16, Stand bei 17)
- Gewinner ermitteln und bekanntgeben
- Ergebnis protokollieren (game_log.json)

☒ Projektanforderungen

Folgende Anforderungen sind an das Projekt gestellt worden:

1. Interaktive Konsolenanwendung
2. Datenvalidierung
3. Dateiverarbeitung

1. Interaktive Konsolenanwendung:

Die Anwendung interagiert mit dem Benutzer durch die Konsole. Benutzer können:

- Menü-Navigation und Spielerentscheidungen
- Hit/Stand-Auswahl während des Spiels
- Neustart des Spiels nach jeder Runde

2. Validierung von Daten:

Folgende Inputs des Benutzers, werden durch die Applikation geprüft, um ein reibungsloses Spielerlebnis für den Benutzer zu garantieren:

- Menüauswahl-Validierung (Historie/Spielen)
 - Gameplay-Entscheidungen (Hit/Stand)
 - Fortsetzungseingabe (Ja/Nein)
-

3. Dateiverarbeitung

Die Applikation verwendet die Datei `game_log.json`, zum Auslesen der Historie und zum Speichern neuer Spielergebnisse.

- **Output file:** `game_log.json`— Die Datei wird beim ersten Spiel generiert. In der Datei befinden sich die letzten Spielergebnisse. Die Spielergebnisse werden immer mit Zeitstempeln versehen.

Implementierung

Technologie

- Python 3.x
- Umgebung: GitHub Codespaces
- Tbd

Repository Struktur

tbd

Wie startet man die Anwendung

tbd

Verwendete Libraries

tbd

Gruppe

| Andri Schwab | | Fisnik Mehmeti |

Contributing

tbd

Lizenz

Dieses Projekt dient nur Bildungszwecken als Teil eines Programmiermoduls. [MIT License](#)