

Aufbau:

- Client.py
 - o GUI.py
 - EventHandler.py
 - GameRender.py
 - GameStateManager.py
 - WidgetManager.py
 - o Network.py

Client.py

Client.py ist die Hauptfunktion des Clients. Hier werden die erforderlichen Funktionen für das GUI aufgerufen und Daten, welche diesen Client betreffen, gespeichert. Es fungiert als Koordinator, der die Kommunikation mit dem Server über *Network.py* initiiert, die Benutzeroberfläche über *GUI.py* steuert und den Spielzustand mittels *GameStateManager.py* verwaltet.

Network.py

In *Network.py* wird das Senden und Empfangen von Daten an und von dem Server umgesetzt. Man übergibt den aktuellen Spielzustand an eine Funktion, um Daten zu senden. Beim Empfangen gibt die Funktion den aktuellen Spielzustand des Servers zurück. Dieses Modul ist für die gesamte Netzwerkkonversation zuständig und stellt sicher, dass der Client mit dem Server synchronisiert ist.

GUI.py

Im *GUI.py* wird das gesamte GUI realisiert. Dort wird zwischen den verschiedenen Menü-Zuständen wie Hauptmenü, Spiel erstellen und schlussendlich dem Spiel gewechselt. Die eigentliche Darstellung wird von *GameRender.py* übernommen. *GUI.py* dient als Schnittstelle für den Benutzer und delegiert die visuelle Ausgabe an *GameRender.py*, während es Benutzereingaben über den *EventHandler.py* verarbeitet.

EventHandler.py

Der Eventhandler wird allein während dem Spiel genutzt. Mit ihm werden die Interaktionen zwischen Spieler und den angezeigten Karten-Elementen erfasst. Die einzelnen Funktionen behandeln, welche Aktion für die aktuell geklickte Kartenart, also Karte im Grid, Ablagestapel oder Nachziehstapel, ausgeführt werden sollen und gibt diese zurück. Die vom *EventHandler.py* erfassten Aktionen werden dann an den

GameStateManager.py weitergegeben, um den Spielzustand entsprechend zu aktualisieren.

GameRender.py

GameRender.py ist verantwortlich für die detaillierte grafische Darstellung des Spiels. Es nimmt die Daten vom *GameStateManager.py* und die Anweisungen von *GUI.py* entgegen, um die verschiedenen Spiel-Elemente wie Karten, Spielfelder und Menüs visuell darzustellen. Dieses Modul kümmert sich um das Zeichnen und Aktualisieren aller visuellen Komponenten auf dem Bildschirm.

GameStateManager.py

Der *GameStateManager.py* wird verwendet, um den aktuellen Spielzustand und alle dafür erforderlichen Daten zu erfassen und zu verarbeiten. In ihm werden Informationen, wie der Zustand des Ablagestapels, die Zulässigkeit von Zügen und weiteren, erfasst und verwertet. Dies beinhaltet auch die Verwaltung von Spielerpunkten, der Zugreihenfolge und der Verteilung von Karten. Die genauen Funktionen werden im Code beschrieben. Dieses Modul ist das zentrale Gehirn des Spiels auf Client-Seite und stellt sicher, dass alle Spielregeln und Zustände korrekt gehandhabt werden.

WidgetManager.py

Da *pygame_widgets* benutzt werden, gibt es diese Funktion, um sich um diese zu kümmern. Widgets werden in Gruppen eingeteilt und erstellt. Über Funktionen können die einzelnen Widgets oder Widgetgruppen versteckt oder angezeigt und deren aktuellen Werte zurückgegeben werden. Dieses Modul erleichtert die Organisation und Manipulation von GUI-Elementen, die mit der *pygame_widgets*-Bibliothek erstellt wurden.

Die genauen Funktionen werden im Code beschrieben.