

Netzwerkprotokoll

CHANGENAME <username>	Anfrage des Clients, sich mit diesem Username mit dem Server zu verbinden
CHANGEOK <username>	Antwort des Servers, dass der Client jetzt verbunden und in der Lobby drin ist unter dem angegebenen Username
CHANGENO	Antwort des Servers, dass dieser Name schon vergeben ist.
CHAT Sender Empfänger isPrivateMessage <message>	Client sendet eine Chatnachricht an Server. Empfänger: Entweder username oder ALL isPrivateMessage (String): Entweder true oder false
GAMESTART	Der Server startet das Spiel.
GIVCARD <cardname>	Server verteilt Karte an Client
TURNSTART	Server teilt dem Client mit, dass sein Zug (10 Sek.) begonnen hat.
TURNEND	Server teilt dem Client mit, dass sein Zug (10 Sek.) fertig ist. Oder: Client teilt dem Server mit, dass er 1 Runde aussetzt.
GETCARD	Client teilt dem Server mit, dass er eine Karte ziehen will.
THROWCARD	Client teilt dem Server mit, dass er eine Karte wegschmeissen (throw away) will.
LOGOUT	Anfrage des Clients an den Server, aus dem Chat auszutreten.

Handler

Server und Clients erstellen ihre Verbindung (Sockets). Dann erstellen sie je ein Handler-Objekt und übergeben ihm den Socket als Parameter. Ab dann verläuft alle Kommunikation via die beiden Handler.

- Die Handler kommunizieren via Netzwerkprotokoll miteinander.
- Die Server-/Client-Programme (z.B. Chat, Spiel, ...) rufen bei ihrem Handler Methoden auf, um mit ihm zu kommunizieren.

Jeder Client hat einen Handler, und der Server hat einen Handler (einen für alle Clients).

Der Server kann bei seinem ServerHandler Methoden aufrufen, um mit den Clients zu kommunizieren. Überladene Methoden:

- Wenn Nachricht an einen einzigen Client geht, gibt er den Username als Parameter mit.
- Wenn die Nachricht an alle Clients geht, gibt er keinen Username mit.

Spiellogik Detailablauf

Turn: Der Zug eines einzelnen Users (10 Sekunden)

Round: Jeder Spieler kommt einmal dran, dann startet die nächste Runde

Match: Sobald jemand 180 Punkte erreicht hat, endet der Match, und es werden Coins vergeben.

Game: Sobald alle genug haben, endet das gesamte Spiel, die Coins werden auf 0 zurückgesetzt.

Starten des Serverprogramms

Server erstellt Socket, auf eingehende Verbindung lauschen auf 8096.

Server erstellt ServerHandler-Objekt und übergibt ihm den Socket als Parameter.

Starten des 1. Clients

Abfrage der IP-Adresse auf Clientsseite.

Client erstellt Socket auf default Port 8096.

Client erstellt ClientHandler-Objekt und übergibt ihm Socket als Parameter.

Server erstellt Thread für Chat.

Client baut GUI der Lobby auf.

Client versucht, sich mit einem Username mit dem Server zu verbinden: CONNE <username>

Server fügt Client zum Spiel und zum Chat hinzu, ggf. unter angepasstem Username, und meldet dem Client zurück, welchen Username er hat: CONOK <username>

Client startet Thread.

Sobald 4 User eingeloggt sind, startet der Server das Spiel mit dem Befehl GAMESTART.

Der Server gibt jedem Client 1 Karte mit GIVCA <cardname>

Server teilt dem 1. Client mit, dass sein Zug begonnen hat: