**Mäxchen-Protokoll**

**Grundlagen**

* Kommunikation zeilenweise per UDP (utf8-kodierte Strings)
* Server öffnet einen bekannten Port
* Clients melden sich auf dem Port als Spieler an und werden ab da für jede Runde angeschrieben
* Clients haben für ihre Antwort ein recht enges Zeitfenster (250 ms)
* Ein Client kann sich als Zuschauer (Spectator) registrieren. Ein Zuschauer kann nicht aktiv an den Runden teilnehmen, erhält aber alle Nachrichten, die an alle Clients geschickt werden.

**Anmelden**

* client->server: REGISTER;name
* client->server: REGISTER\_SPECTATOR;name

Falls name gültig und neu ist, oder die Registrierung unter demselben Namen zuletzt von derselben IP kam:

* Server kommuniziert mit dem Client ab jetzt über die Ursprungs-IP und den Ursprungs-Port der Register-Nachricht.
* server->client: REGISTERED

Ansonsten:

* server->client: REJECTED

Kriterien für gültige Spielernamen:

* enthalten keinen whitespace
* enthalten keine Doppelpunkte, Semikolons oder Kommas
* sind maximal 20 Zeichen lang

**Start einer Spielrunde**

1. server->clients: ROUND STARTING;token
2. client->server: JOIN;token

Falls mindestens ein Spieler teilnehmen will:

* Die teilnehmenden Spieler werden vom Server in eine zufällige Reihenfolge gebracht
* server->clients: ROUND STARTED;rundennummer;spielernamen  - wobei spielernamen eine kommagetrennte List der Mitspieler ist, in der Reihenfolge, in der diese Runde gespielt wird.

Ansonsten:

* server->clients: ROUND CANCELED;NO\_PLAYERS - woraufhin eine neue Runde gestartet wird.
* Runden mit nur einem Spieler werden direkt nach dem Start mit ROUND CANCELED;ONLY\_ONE\_PLAYER abgebrochen.

**Ablauf einer Spielrunde**

***Ankündigungen***

Reihum:

1. server->client: YOUR TURN;token
2. client->server: command;token (wobei command eins der folgenden ist: ROLL, SEE)

Bei ROLL:

* server->clients: PLAYER ROLLS;name
* server->client: ROLLED;dice;token
* client->server: ANNOUNCE;dice';token
* server->clients: ANNOUNCED;name;dice

Falls Mäxchen angesagt wurde, wird sofort aufgedeckt. Wenn tatsächlich Mäxchen gewürfelt wurde, verlieren alle anderen Spieler, ansonsten der ansagende.

* server -> clients: PLAYER LOST;names;reason (wobei names eine kommagetrennte liste ist)
* Bei SEE:
* Server überprüft, ob zuletzt angesagte Würfel okay sind und bestimmt, wer verloren hat
* server->clients: PLAYER WANTS TO SEE;name
* server->clients: ACTUAL DICE;dice
* server->clients: PLAYER LOST;name;reason

***Rundenende***

Wann immer ein Spieler nicht rechtzeitig antwortet oder etwas völlig falsch macht:

* server->clients: PLAYER LOST;name;reason

Nach Ende einer Runde:

* server->clients: SCORE;spielerpunkte\* (wobei spielerpunkte eine kommagetrennte Liste von Einträgen in der Form name:punkte ist)

**Gründe, warum ein Spieler verliert**

* SEE\_BEFORE\_FIRST\_ROLL: Spieler wollte sehen, war aber als erster am Zug (es gab also noch keine Ansage vorher)
* LIED\_ABOUT\_MIA: Spieler hat Mäxchen angesagt, ohne Mäxchen zu haben
* ANNOUNCED\_LOSING\_DICE: Spieler hat zu niedrige Würfel angesagt
* DID\_NOT\_ANNOUNCE: Spieler hat nicht (rechtzeitig) angesagt, was gewürfelt wurde
* DID\_NOT\_TAKE\_TURN: Spieler hat nicht (rechtzeitig) einen Zug gemacht
* INVALID\_TURN: Spieler hat einen ungültigen Zug gemacht
* SEE\_FAILED: Spieler wollte sehen, Ansage des vorhergehenden Spielers war aber richtig
* CAUGHT\_BLUFFING: Spieler hat mehr angesagt als er hatte, und der nachfolgende Spieler wollte sehen
* MIA: Es wurde Mäxchen aufgedeckt