## Client - Server Kommunikation:

**Benutzerhandbuch:** nur clientseitig

**Welcome – Verbinden:**

Client: Username-Vorschlag basierend auf System Username

Client → Server: Username

Server: check username in Database, falls doppelt, hänge kleinste Zahl hintendran (nochmal testen bis nicht vorhanden), schreibe User mit IP in Datenbank

Server → Client: Send Back chosen Username

**Lobby - Username wechseln:**

Client → Server: Username

Server: check username in Database, falls doppelt, hänge kleinste Zahl hintendran (nochmal testen bis nicht vorhanden), suche User anhand von IP / ID und speichere neuen Username

Server → Client: Send Back chosen Username

*Server hat letztes Wort bei Vergabe des Usernamens (?), User kann zwar Username nochmals wechseln, wir aber nicht gefragt ob er hinten eine Zahl dran will (?)*

**Lobby - Lobby verlassen:**

* abmeldevorgang
* … ?

**Lobby - Gruppe Info:**

Client → Server: gib Gruppeninfo zu Gruppe1  
Server: sucht Gruppe anhand von Name oder ID ??? siehe Gruppename check (duplicates)

Server → Client: sende Liste der Mitglieder der Gruppe

**Lobby - reload:**

Client → Server: ask for reload

Server: holt alle open games aus der DB + Zahl der aktuell angemeldeten Spieler

Server → Client: sende alle aktuell offenen Games + Zahl der aktuell angemeldeten Spieler zurück.

**Lobby - Spiel erstellen**

Client: Popup Window zu Spiel erstellen, warte auf Usereingabe

**Erstellen:**

Client → Server: Gruppenname, Anzahl Spieler

Server: erstellt Spiel

Server → Client: ok

Server → all Clients : , sendet aktualisierte Liste von offenen Spielen an alle verbundenen Clients (Lobby updaten)

Client: weiter zu Warten auf Spielbeginn

**Lobby - Beitreten**

Client: User gibt Spiel-ID ein

Client → Server: sende Anfrage zum Beitreten

Server: sucht Spiel-ID (falls nicht vorhanden Abbruch), check ob voll, falls nicht voll

Server → Client: ok oder abbruch

Client: falls ok, weiter zu Warten auf Spielbeginn

**Lobby - Show All Games**

Client → Server: Abfrage alle Spiele

Server → Client: Liste aller Spiele

Client: Darstellen

**Warten auf Spielbeginn:**

Client → Server: beitreten (siehe entweder lobby beitreten oder spiel mit id beitreten)

Server: nimmt Client in Playerliste auf (für Multicast)

Server → Client: Infos zur Gruppe + vergangene Aktivitäten

Client: Socket aufsetzen um Chatnachrichten dieser Gruppe zu empfangen

**Während den Spielphasen:**

Server→ Client : Farben zuordnen, alle Client Skala anzeigen,Radkarten verteilen

Server → Client : änderung der Radkarten im Stapel und Zielkarten

Server → Client : verteilte Karten individuell angezeigt werden

Client : Darstellen

Client → Server : chat

Server → Client : chat Message weiterleiten

Client → Server : austreten

Server : user name von der List des jetzigen Spiels entfernen

**Spielphase Vorbereitung**

Server → Client : random Zielkarten senden,auch in der Spielphase ”Während des Spiels”

Client → Server : ausgewählte Zielkarten,,auch in der Spielphase ”Während des Spiels”

**Während des Spiels:**

Server : ständig Anzahl der Radkarten von allen Clients prüfen

Server : Punkte rechnen für jeder Spieler

Server → all Clients : Reminder : Wer ist daran

Client → Server : Radstreckenvorlage drücken

Server → Client : Warnung /ohne Warnung /random Farbe der Karten ausgeben

Client → Server : Radstrecken Vorlagen bauen

Server →all Clients : Radstrecken Vorlagen aktualisieren

**Endphase**

Server → Client : Rangliste und individuelle Achievements aufzeigen

Client → Server : safe logout

Server : Programm beenden

## Client-Server-Protokoll

Folgende Liste dient nur als Überblick des Protokolls. In diesem Abschnitt muss aber das Protokoll stehen.

* Spielverlauf- und Zustand (regelkonforme Züge etc.) von Server kontrolliert
* Verwaltung meherer Spielinstanzen durch Server
* Verwaltung von Liste offener Spiele durch Server
* Server sendet Client Ping-Signal und Client sendet Server Pong-Signal, um Verbindungsverluste zu erkennen und handzuhaben.
* Broadcast hat eigenen Namen
* eindeutige Definition jedes Befehls
* Protokoll erweiterbar
* Seite 9 auf den Slides

## Ideen für Klassen

Player / Client:

- interne ID

- nickname

- IP

- Farbe

Chat:

- inwiefern werden Chatnachrichten in der DB gespeichert

- nach Spiel nicht mehr wichtig

- Empfang sicherstellen ... blabla

Spielzug:

- Text der im Feld "Aktivitäten" angezeigt wird

- Aktivitätsart (Zielkarte gezogen, Radkarte gezogen, Strecke gebaut) --> entsprechen Klassen mit wichtigen Infos

Strecke

- Player

- Farbe

Game

- PlayerList

- Anzahl Spieler (wie viele Spieler definiert)

- Gruppenname

- Status (open, ongoing, finished)