

HSD-Bingo

Kommunikationsdokument

V 0.3 / 13.12.2020

FH Hagenberg
ISE-Projekt WS20/21

Dokument-Information

Gruppenname: Gruppe 2

Autor: Lucas Drack / Daniel Kogler / Michael Reiter

Gruppenmitglieder: S2010567003 – Drack Lucas – Server
S2010567017 – Kogler Daniel – Desktopclient
S2010567006 – Reiter Michael – Webclient

Datei: ISE_G2_Kommunikation.docx

Version: 0.3

Versionsdatum: 13.12.2020

Dokumenthistorie

| Version | Datum | Geänderter Inhalt | Bemerkungen |
|---------|------------|--|---|
| 0.1 | 24.11.2020 | Erste Version | Beinhaltet Definitionen aller Kommandos für die Basisversion von HSD-Bingo. Kommunikation basiert auf HTTP. |
| 0.2 | 04.12.2020 | Spielstatus abfragen, Bestenliste abfragen | Spielstatus abfragen mittels LVA Name, Bestenliste Abfragen hinzugefügt, ISE1 zu ISE7 geändert |
| 0.3 | 13.12.2020 | Disconnect-Kommando festgelegt | Korrektur bei /bingo |
| | | | |

Inhalt

| | |
|---|----|
| Dokument-Information..... | 2 |
| Dokumenthistorie..... | 2 |
| Kommunikationsschnittstelle | 4 |
| Lobby-Liste abfragen | 4 |
| Einer Lobby beitreten | 5 |
| Liste aller Spieler mit Spielstand und Zustand anfragen..... | 6 |
| Lobby-Status anfragen..... | 7 |
| Spiel Starten..... | 8 |
| Eigenes Spielfeld (5x5 Felder) anfragen..... | 9 |
| Ein Feld anklicken und Auswahl treffen..... | 10 |
| Spielstatus anfragen | 11 |
| Bestenliste abfragen | 12 |
| Sequenzdiagramm für den generellen Ablauf: Spielbeitritt – Spiel – Spielende..... | 14 |

Kommunikationsschnittstelle

Die Gesamte Kommunikation zwischen den Clients und dem Server findet über HTTP Requests statt. Verbunden wird sich auf den Port 8080. Der Server schickt die Daten im JSON Format an die Clients aus.

Allgemeine Struktur:

POST http://<Server-IP-Adresse>:8080/<Spielkommando>

content-type: application/json

[{<Name>:<Wert>,...}, {...},...]

Rückgabe:

[{<Name>:<Wert>,...}, {...},...]

bzw.:

GET http://<Server-IP-Adresse>:<Spielkommando>

Rückgabe:

[{<Name>:<Wert>,...}, {...},...]

Lobby-Liste abfragen

GET http://193.170.192.202:8080/getlobbies

Rückgabe:

[{"name":<Name der LVA>,"users":<Anzahl der User>}, ... , ...]

| Bezeichner | Beispiele | Beschreibung |
|-------------------|-------------------|----------------------------------|
| < Name der LVA > | ANA1, CHD4, ISE7, | Name der Lehrveranstaltung |
| <Anzahl der User> | 0, 1, 2, 3, 4, 5 | Anzahl der beigetretenen Spieler |

Beschreibung:

Dieses Kommando dient dem Abfragen der verfügbaren Lobbies. Dem Kommando müssen keine Attribute mitgegeben werden. Der Server antwortet mit einer Liste der verfügbaren Lobbies.

Einer Lobby beitreten

POST <http://193.170.192.202:8080/joinlobby>

{"lobby": <Lobbyname>, "name": <Spielername>}

| Bezeichner | Beispiele | Beschreibung |
|---------------|---------------------------------|---------------------------------------|
| <Lobbyname> | "ISE7", "SIS3", "RTO7" | Name der Lobby (Lehrveranstaltung) |
| <Spielername> | "Mario", "Drack", "Kogler", ... | Name des Spielers |

Rückgabe:

{"UID":<Spieler ID>, "master":<Master-Status>}

| Bezeichner | Beispiele | Beschreibung |
|-----------------|---------------------|---|
| <Spieler ID> | 4711, 256643, 15145 | Identifikationsnummer des Spielers |
| <Master-Status> | true, false | Beschreibt, ob der Spieler "Master" der aktuellen Lobby ist und somit als einziger das Spiel starten kann |

Beschreibung:

Der Spieler will mit seinem Namen einer Lobby beitreten. Der Server antwortet mit einer eindeutigen Spieler-ID, sowie einem Status, welches angibt ob der Spieler "Master" der Lobby ist (in der Regel der erste Spieler in der Lobby). Ist der Spielername ungültig (bereits vorhanden) antwortet der Server mit Spieler-ID:0 und somit wird der Spielername abgelehnt und der Spieler muss versuchen, mit einem anderen Namen beizutreten.

Liste aller Spieler mit Spielstand und Zustand anfragen

GET <http://193.170.192.202:8080/getplayers>

{ "lobby": <Lobbyname> }

| Bezeichner | Beispiele | Beschreibung |
|-------------|------------------------|---------------------------------------|
| <Lobbyname> | "ISE7", "SIS3", "RTO7" | Name der Lobby (Lehrveranstaltung) |

Rückgabe:

[{"name":<Spielername>, "score":<Spielstand>, "status":<Spielers-Zustand>}, {...}]

| Bezeichner | Beispiele | Beschreibung |
|-------------------|---------------------------------|-----------------------------------|
| <Spielername> | "Mario", "Drack", "Kogler", ... | Name des Spielers |
| <Spielstand> | 0, 1, 2, 3, 4, 5, .. | Länge der längsten Wortkette |
| <Spieler-Zustand> | "ready", "waiting" | Aktueller Zustand des Spielers |

Beschreibung:

Mit diesem Kommando können die Spieler abgefragt werden, die sich im Warteraum befinden. Weiters wird übertragen, ob der Spieler schon bereit ist, um zu spielen. Dieses Kommando kann auch während des Spiels benutzt werden, um weitere Informationen über die anderen Spieler zu erhalten, wie zum Beispiel die aktuell längste Wortkette.

Lobby-Status anfragen

GET http://193.170.192.202:8080/getlobbystatus

{ "lobby": <Lobbyname> }

Rückgabe:

{ "gametime": <Spielzeit> }

| Bezeichner | Beispiele | Beschreibung |
|-------------|------------------------|--|
| <Lobbyname> | "ISE1", "SIS3", "RTO1" | Name einer Lobby |
| <Spielzeit> | 0, 34, 666, ... | Abgelaufene Zeit seit dem Spielstart in Sekunden |

Beschreibung:

Dient zum Anzeigen der Verfügbaren Lobbys. Der Server kennt die Lobbys und schickt diese beim Starten des Spiels an die Clients aus. Die Lobbys werden dann zusammen mit der Startzeit und der Zahl der aktiven Spieler am Startbildschirm angezeigt.

Spiel Starten

POST http://193.170.192.202:8080/start

{"UID": <Spieler ID>}

| Bezeichner | Beispiele | Beschreibung |
|--------------|---------------------|------------------------------------|
| <Spieler ID> | 4711, 256643, 15145 | Identifikationsnummer des Spielers |

Beschreibung:

Nur der "Master" kann das Spiel starten. Dies wird erreicht, indem der Client den "Starten"-Knopf nur dann anzeigt, wenn er durch Kommando joinlobby den "master"-Bezeichner erhalten hat. Ruft ein Spieler, der nicht Master ist, diesen Request auf, wird es vom Server als Ready gewertet.

Eigenes Spielfeld (5x5 Felder) anfragen

GET <http://193.170.192.202:8080/getboard>

{"UID": <Spieler ID>}

| Bezeichner | Beispiele | Beschreibung |
|--------------|---------------------|------------------------------------|
| <Spieler ID> | 4711, 256643, 15145 | Identifikationsnummer des Spielers |

Rückgabe:

[<Begriff>, <Begriff>, <Begriff> , ...]

| Bezeichner | Beispiele | Beschreibung |
|------------|--|---|
| <Begriff> | "urlangsam", "sinngemäß", "performant", ... | Begriff für das entsprechende Spielfeld |

Beschreibung:

Begriffe werden Reihe für Reihe aufgelistet. Beginnend mit Reihe 0 / Spalte 0 (links oben) bis Reihe 5 / Spalte 5 (rechts unten).

Ein Feld anklicken und Auswahl treffen

POST <http://193.170.192.202:8080/makeselection>

{"UID": < Benutzer-ID>, "x": <Zeilennummer>, "y": <Spaltennummer>}

| Bezeichner | Beispiele | Beschreibung |
|-----------------|-------------|-----------------------------------|
| < Benutzer-ID > | 4711,42,123 | Benutzer Identifikationsnummer |
| <Zeilennummer> | 0,1,2,3,4 | Zeilennummer |
| <Spaltennummer> | 0,1,2,3,4 | Spaltennummer |

Beschreibung

Wenn der Client ein Kästchen auf seinem Spielfeld anklickt, schickt er die Koordinaten des Kästchens und seine ID zum Server. Das Kästchen in der linken Spalte und der obersten Zeile trägt dabei die Koordinaten (0,0).

Spielstatus anfragen

GET http://193.170.192.202:8080/bingo

{ "lobby": <Lobby> }

| Bezeichner | Beispiele | Beschreibung |
|------------|------------------|------------------|
| < Lobby > | ISE7, RTO7, SIS3 | Name einer Lobby |

Rückgabe

{ "winner": null } wenn es noch keinen Gewinner gibt

{ "winner": <Spielername> } wenn <Spielername> das Spiel gewonnen hat

| Bezeichner | Beispiele | Beschreibung |
|---------------|-----------------------------|--------------|
| <Spielername> | "Mario", "Drack", "Kogler", | Spielername |

Beschreibung

Der Client benutzt dieses Kommando, um den Spielstatus zu pollen. Sobald der Server registriert, dass ein Spieler gewonnen hat, werden hiermit alle Clients benachrichtigt. Diese zeigen dann den Gewinner- bzw. Verliererbildschirm an, je nachdem ob der eigene Name als Siegername erkannt wurde.

Bestenliste abfragen

GET <http://193.170.192.202:8080/highscores>

Rückgabe

```
{  
  "bestPlayers": [{ "Name": <Spielername>, "Score": <score> }, { "Name":  
    <Spielername>, "Score": <score> }, ... ],  
  "quickestLobbies": [{ "lobby": <Lobbyname>, "gametime": <Zeit> },  
    { "lobby": <Lobbyname>, "gametime": <Zeit> }, ... ]  
}
```

| Bezeichner | Beispiele | Beschreibung |
|---------------|-----------------------------|------------------------------------|
| <Spielername> | "Mario", "Drack", "Kogler", | Spielername |
| <score> | 107,42,3 | Anzahl der jemals gewonnenen Spiel |
| <Lobbyname> | "ISE7", "RTO7", "SIS3" | Name der LVA |
| <Zeit> | 12345, 400, 50 | Zeit in Sekunden |

Beschreibung

Die 10 Spieler mit den meisten Gesamtsiegen unabhängig von der LVA werden auf Anfrage vom Server geliefert. Auch die 10 schnellsten Zeiten, in denen ein Spiel gewonnen wurde, werden mitgeschickt.

Verbindung (freiwillig) abbrechen

POST <http://193.170.192.202:8080/disconnect>

{"UID": < Benutzer-ID>, "lobby": <Lobbyname>}

| Bezeichner | Beispiele | Beschreibung |
|-----------------|------------------------|-----------------------------------|
| < Benutzer-ID > | 4711,42,123 | Benutzer Identifikationsnummer |
| <Lobbyname> | "ISE7", "RTO7", "SIS3" | Name der LVA |

Beschreibung

Mit diesem Kommando können die Clients ihre Verbindung zum Spiel trennen.

Egal, in welchem Zustand die Spielelobby ist (waiting/running/finished), wird der Zustand des Spielers auf disconnected gesetzt. Sobald alle Spieler einer Lobby die Verbindung getrennt haben, wird die Lobby resettiert und ein neues Spiel kann beginnen.

Sequenzdiagramm für den generellen Ablauf: Spielbeitritt – Spiel – Spielende

