Server-Protokoll (v0.9)

Patrick Schäfer
patrick.schaefer@hu-berlin.de

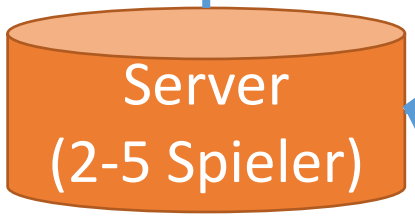
Semesterprojekt: Implementierung eines Brettspiels, WS 18/19

Anforderungskatalog

Playback

JSON

JSON



1a) Meilenstein
Observer-GUI

1b) Meilenstein
Single-Player

2. Meilenstein
KI-Spieler /
Wettbewerb

JSON …

- Wird zur Übertragung und zum Speichern strukturierter Daten verwendet
- Es existieren folgende Datentypen:
  1. Null-Werte (null)
  2. Bool‘sche Werte (false und true)
  3. Zahlen
  4. Zeichenketten
  5. Arrays
     - beginnt mit []
     - Beispiel: [1,2,3,4,5,6]
  6. Objekte (entspricht einer HashMap in Java)
     - beginnen mit {}

     Einträge haben die Form Schlüssel:Wert
     - Beispiel: {"Name": "Mustermann", "Vorname": "Max"}

Beispiel


```
{"type":"Info",
 "player":"Red",
 "success":true,
 "turnType":"BoardState",
 "drawnDestinationTickets":[],
 "drawnPassengerCars":[2,1,0,0,0,0,0,0,1],
 "faceUpPassengerCarDeck":["Green","Yellow","Red","Purple","Yellow"],
 "leftPassengerCars":45,
 "toBeClaimedDestinationTickets":[
    {"city1":"Vancouver","city2":"Montreal","points":20},
    {"city1":"SanFrancisco","city2":"Atlanta","points":17},
    {"city1":"Winnipeg","city2":"LittleRock","points":11}],
}
```

JSON-Protokoll

- Server Nachrichten werden ...
  - mittels JSON serialisiert
  - über einen Socket verschickt oder in eine Log-Datei geschrieben
  - müssen von euch deserialisert werden.

- Der Server sendet zwei Arten von Nachrichten:
  - **Info**: Züge anderer Spieler
  - **Request**: Benachrichtig Spieler-Client, welcher Zug von ihm erwartet wird

Server-Nachrichten

- Eine Nachricht vom Server enthält die folgenden Felder:

Type	: Info oder Request
turnType	: Antwort auf euren Zug
success	: Die Anfrage war erfolgreich (nur wenn Type = Info)
player	: Der Spieler von dem der Zug gemacht wurde
errorCode	: Ein Fehler-code (nur wenn success = false)

Server-Nachrichten

- Beispiele für Server-Nachrichten:
  - Request
    {"TYPE":"REQUEST","PLAYER":"BLUE","TURNTYPE":"**JOIN**"}

  - INFO
    {"TYPE":"INFO","PLAYER":"BLUE",
    "TURNTYPE":"**JOIN**","SUCCESS":TRUE, "PLAYERNAME":"**PATRICK**"}

- Erstmal betrachten wir nur Info-Nachrichten
- Request benötigen wir für die KI und den Single-Player-Modus

Spielmaterial und Datentypen

- Die Wagenkartenfarben:

Purple, White, Blue, Yellow, Orange, Black, Red, Green, Rainbow

- Die Ziele:

Chicago, LosAngeles, Montreal, Atlanta, Calgary, Denver, ElPaso, Houston, LasVegas, NewOrleans, OklahomaCity, Phoenix, Portland, SaintLouis, SaultStMarie, Washington, SanFrancisco, Toronto, Boston, Charleston, Dallas, Duluth, Helena, KansasCity, LittleRock, Miami, Nashville, NewYork, Omaha, Pittsburgh, Raleigh, SaltLakeCity, SantaFe, Seattle, Vancouver, Winnipeg

- Zielkarten

{"CITY1":"A","CITY2":"B","POINTS": 20}

- Die Spielerfarben

Blue, Red, Green, Yellow, Black

Das Spielbrett

- Eine Nachricht vom Typ “***BoardState***“ gibt das aktuelle Spielbrett aus Sicht des Spielers aus

- Allgemeine Informationen:
  faceUpPassengerCarDeck : Array von Wagenkarten
  topdownPassengerCarDeckCount : Int
  destinationTicketsCount : Int
  finalTurn : Boolean

- Handkarten:
  leftPassengerCars : Int
  drawnPassengerCars : Array von Wagenkarten
  drawnDestinationTickets : Array von Zielkarten
  toBeClaimedDestinationTickets : Array von Zielkarten
  ownRoutes : Array von Routen

Das Spielbrett

- Beispiel:


```
{"type":"Info","player":"Red","success":true,"turnType":"BoardState",
 "destinationTicketsCount":20,
 "drawnDestinationTickets":[],
 "drawnPassengerCars":[2,1,0,0,0,0,0,0,1],
 "faceUpPassengerCarDeck":["Green","Yellow","Red","Purple","Yellow"],
 "finalTurn":false,
 "leftPassengerCars":45,
 "ownRoutes":[],
 "toBeClaimedDestinationTickets":[
    {"city1":"Vancouver","city2":"Montreal","points":20},
    {"city1":"SanFrancisco","city2":"Atlanta","points":17},
    {"city1":"Winnipeg","city2":"LittleRock","points":11}],
 "topdownPassengerCarDeckCount":89
}
```

Ein Spielzug

- Für jede Spieler-Aktion existiert eine eigene Nachricht
  1. Spiel beigetreten
     Join
  2. Wagenkarte nehmen (2-phasig)
     DrawPassengerCars
  3. Strecke bauen
     ClaimRoute
  4. Zielkarte ziehen (2-phasig)
     DrawDestinationTickets
     ClaimDestinationTickets
  5. Spielende
     FinalScore

1. Spiel beigetreten

- Eine Nachricht vom "***Join***" gibt an, dass ein Client beigetreten ist
- Ein Client kann als „Observer" oder als „Spieler" beitreten
- Informationen:
  - **playerName :** String
  - **clientType** : String („Observer" oder „Spieler")

1. Spiel beigetreten

- Beispiel:


```
{
 "TYPE":"INFO",
 "PLAYER":"BLUE",
 "SUCCESS":TRUE,
 "TURNTYPE":"JOIN",
 "CLIENTTYPE":"PLAYER",
 "PLAYERNAME":"B“
}
```

1. Wagenkarte nehmen

- Eine Nachricht vom Typ "***DrawPassengerCars***" gibt an, dass ein Spieler eine Karte gezogen hat
- Die Karte kann vom verdeckten Stapel (**hiddenDeck**) oder von den offen ausliegenden Karten gezogen worden sein
- Informationen:

  drawnCard **:** Farbe

  faceUpPassengerCarDeck : Array von Wagenkarten

  hiddenDeck : boolean

- Die Wagenkartenfarben:
  Purple, White, Blue, Yellow, Orange, Black, Red, Green, Rainbow

Wagenkarten

1. Wagenkarte nehmen

- Beispiel:


```
{       "TYPE":"INFO","PLAYER":"RED","SUCCESS":TRUE,
        "TURNTYPE":"DRAWPASSENGERCARS",
        "DRAWNCARD":"ORANGE",
        "FACEUPPASSENGERCARDECK":
                ["WHITE","BLACK","BLACK","PURPLE","WHITE"],
        "HIDDENDECK":FALSE
        }
```

2. Strecke bauen

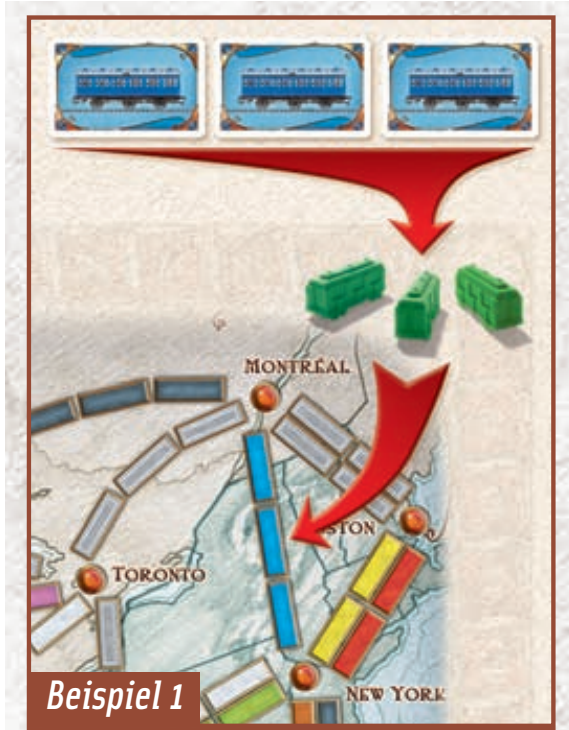
- Eine Nachricht vom Typ "***ClaimRoute***" gibt an, dass ein Spieler eine Strecke gebaut hat
- Es wird angegeben, mit welchen Wagenkarten bezahlt wurde
- Informationen:

  d1 *: Ziel 1*

  d2 *: Ziel 2*

  passengerCarColors ***:*** *Array von* Wagenkarten

- Die Wagenkartenfarben:
  Purple, White, Blue, Yellow, Orange, Black, Red, Green, Rainbow



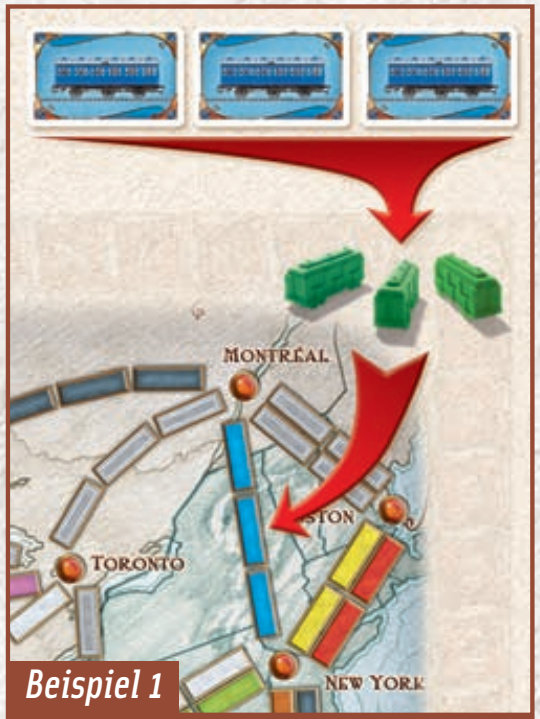
Beispiel 1

Um die Strecke von Montreal nach New York zu befahren, benötigt der Spieler ein Set aus drei blauen Personenwagen-Karten.

2. Strecke bauen

- Beispiel:


```
{
 "TYPE":"INFO","PLAYER":"BLUE","SUCCESS":TRUE,
 "TURNTYPE":"CLAIMROUTE",
 "D1":"MONTREAL",
 "D2":"NEWYORK",
 "PASSENGERCARCOLORS":["BLUE","BLUE","BLUE"]
}
```

Beispiel 1

Um die Strecke von Montreal nach New York zu befahren, benötigt der Spieler ein Set aus drei blauen Personenwagen-Karten.

3. Zielkarte ziehen

- Zielkarten werden in zwei Phasen gezogen:
  DrawDestinationTickets *und* ***ClaimDestinationTickets***
- Beide Nachrichten enthalten die Information, welche Karten gezogen wurden

- Information:

  drawnCards **:** Array von Zielkarten

- Zielkarten-Format
  {"CITY1":"A","CITY2":"B","POINTS": 20}

3. Zielkarte ziehen

- Beispiel:


```
{       "TYPE":"INFO","PLAYER":"BLUE","SUCCESS":TRUE,
        "TURNTYPE":"DRAWDESTINATIONTICKETS",
        "DRAWNCARDS":[ {"CITY1":"KANSASCITY","CITY2":"HOUSTON","POINTS":5},
                       {"CITY1":"LOSANGELES","CITY2":"MIAMI","POINTS":20},
                       {"CITY1":"WINNIPEG","CITY2":"HOUSTON","POINTS":12}]}

{       "TYPE":"INFO","PLAYER":"BLUE","SUCCESS":TRUE,
        "TURNTYPE":"CLAIMDESTINATIONTICKETS",
        "DRAWNCARDS":[{"CITY1":"WINNIPEG","CITY2":"HOUSTON","POINTS":12}]}
```

30 Zielkarten

Spielende

- Eine Nachricht vom Typ "***FinalScore***" gibt den finalen Punktestand aus

- Information:
  - **totalScore**: int
  - **scorePassengerCars**: int
  - **longestRouteLength**: int
  - **winner**: boolean
  - **longestRoute**: boolean
  - **claimedTickets**: Array von Zielkarten
  - **nonClaimedTickets**: Array von Zielkarten

Spielende

- Beispiel

{"type":"Info","player":"Red","success":true,
"turnType":"FinalScore",
"claimedTickets":[],
"longestRoute":false,
"longestRouteLength":3,
"name":"B",
"nonClaimedTickets":
[{"city1":"Denver","city2":"Pittsburgh","points":11},
{"city1":"Helena","city2":"LosAngeles","points":8},
...],
"scorePassengerCars":54,
"totalScore":-64,
"winner":true}