Dokumentation der Komponenten

Unsere Architektur basiert im Groben auf drei Komponenten: Dem *Session*-Service, dem *Chat*-Service und dem *User*-Service. Der *Chat*-Service bildet die Basis Komponente, welche den eigentlichen Chatflow zur Verfügung stellt. Der *User*-Service verwaltet Nutzer Accounts und der *Session*-Service überwacht aktive Spiele.

Chat

Die Chatkomponente interpretiert die abgelegten Entscheidungsbäume und liefert auf Basis eines Inputs in Form einer Textnachricht eine entsprechende Antwort.

Chatflows
Ein Chatflow wird durch seinen Entscheidungsbaum modelliert. Der Entscheidungsbaum wird im JSON-Format in einer Datei festgehalten und durch einen Interpreter in einen Chatflow übersetzt. Die Chatkomponente vermag von einem Ausgangspunkt und einem Input die nächsten Schritte abzuleiten. Sie funktioniert ähnlich einer State-Machine.

Beispiel: Entscheidungsbaum in JSON:

```
{
    "id": "0",
    "type": "question",
    "question": "Hey, ich kenne dich nicht, aber deine Nummer stand auf dem Klo. Kannst du mir helfen???",
    "choices": [
        {
            "nextNode": "3",
            "pattern" : "((?i)ja|ok|klar|jo|yo|natürlich|auf jeden fall)+.*"
        },
        {
            "nextNode": "1",
            "pattern" : "((?i)nein|ne|nope|auf keinen fall)+.*"
        }
    ]
},
```

Nodes
Unser *Node-Interface* wird von vier verschiedenen Nodes implementiert (Message-Node, Death-Node, End-Node und Question-Node).

Beispiel: Question-Node:

```
private QuestionNode(
  String question,
  Status status,
  Set<Choice> choices
) {
  this.question = question;
  this.status = new AtomicReference<>(status);
  this.choices = choices;
}
```

Die Question-Node setzt die zentrale Funktionalität um, die Eingaben des Spielers entsprechend zu interpretieren. Dazu enthält sie ein Set von *Choices.* Jede Choice hat ein

Datenpaar aus *regulärem Ausdruck* und der folgenden Node. Die Choice hat eine *test*-Methode implementiert, welche prüft ob die eingegangene Message auf einen der regulären Ausdrücke passt.

User

Die User Komponente persistiert die Spielstände aller Spieler. Zu einem User können später entsprechende Statistiken gespeichert werden, also seine gespielten und seine gewonnenen Spiele.
Das *User-DAO* entkoppelt die Persistenz von der Logik, der *User-Service* verwaltet diese.

Session

Die Session Komponente managed alle ein- und ausgehenden Nachrichten. Sobald der User einer Chatnachricht schickt, wird diese einer aktiven Session zugeordnet. Die Session kommuniziert eng mit der Chat Komponente und reicht den User Input in die State Machine des entsprechenden Chatflows und legt den Output wieder als Chatnachricht fest und schickt sie zurück an den User.