

Verwendete Technologien beim Server:

- Sprache: Java SE Version 8
- UTF8 Zeichenkodierung, Kommunikation über ein WebSocket (RFC 6455), Serialisierung über JSON
- Entwicklungsumgebung: eclipse Version 2019-06
- Verwendete Bibliotheken und Frameworks:
 - ➔ JRE System Library
 - ➔ Junit Version 5, Lizenz: <https://mvnrepository.com/artifact/junit/junit>
 - ➔ com.google.code.gson Version 2.8.6, Lizenz: <https://mvnrepository.com/artifact/com.google.code.gson/gson/2.8.6>
 - ➔ javax.websocket Version 1.0, Lizenz: <https://mvnrepository.com/artifact/javax.websocket/javax.websocket-api/1.1>
 - ➔ org.glassfish.tyrus (tyrus-server) Version 1.1
 - ➔ org.glassfish.tyrus (tyrus-client) Version 1.1
 - ➔ org.glassfish.tyrus (tyrus-container-grizzly) Version 1.1Lizenzen: <https://mvnrepository.com/artifact/org.glassfish.tyrus>

Tests bzw. Qualitätsmaßnahmen:

Der Server wurde in der Messe eingekauft. Unit Tests waren beim Kauf der Komponente enthalten. Unit Tests darüber hinaus wurden keine geschrieben. Das Verhalten des Servers wurde bei Laufversuchen genau beobachtet und bei Fehlern wurde dieser ausgebessert. Beim Ausbessern wurden jeweilige Code Fragmente gedebugged. Da das Debuggen, also Bereinigen von Fehler auch eine Qualitätsmaßnahme ist, werden alle beobachteten Server Fehler, die gefixt wurden hier aufgelistet:

- Der Server war ursprünglich mit einem TCP Socket implementiert worden. Der Server wurde so umgeschrieben, so dass dieser ein multithreaded WebSocket implementiert. Dabei wurde die Netzwerkschnittstelle des Servers reviewed.
- Tresore waren alle mit 1 durchnummeriert
- Initialisierung der Game Chips auf den Roulette Tisch Feldern fehlte
- Nach einer GamePause Message hatte der Server einen Zustandsfehler: Er befand sich fortan im GamePause Zustand.
- GameStatus Message wurde am Anfang der Runde nicht abgeschickt
- Die Bewegungspunkte und Aktionspunkte des active Characters wurden nach dem Senden der GameStatus Message aktualisiert (diese waren nicht in der Nachricht enthalten)
- Wahlphase wurde falsch implementiert: Höchst- und Mindestanzahl wurden während der Generierung von Items nicht mitberücksichtigt #1
- Spieler bzw Zuschauer konnten sich mit dem gleichen Namen registrieren #2
- Cat Action Zielkoordinaten und State Zielkoordinaten stimmten nicht überein
- Mehr als ein Diamanthalband kann im Umlauf sein
- Bei Cat Action werden keine Plätze getauscht, falls sich die Katze auf das Feld eines Charakters zubewegt
- IP Punkte werden nicht kalkuliert, wenn die Siegbedingung eintritt
- „Übergabe“ des Diamanthalbandes an die Katze ist nicht konform mit dem Lastenheft
- Katze kann das Diamanthalband nicht selber aufnehmen, falls sie das Feld mit Diamanthalband betritt (fehlender Methodenaufruf)
- Sichtlinie wurde falsch implementiert: Felder, bei denen „nur“ die Ecken geschnitten werden, blockieren auch die Sichtlinie!
- Es können Geheimnisse zu Tresoren erhalten werden, die nicht existieren!

#1 wurde manuell getestet:

- Wahl von 6 Gadgets
- Wahl von 4 Agenten und
- ohne Erreichung der Höchstgrenze

#2 wurde manuell getestet:

- 2 Spieler mit dem gleichen Namen
- Zuschauer mit dem gleichen Namen
- Spieler und Zuschauer mit dem gleichen Namen
- Registrierung während des Spiels und vor dem Spiel
- Freiwerden von Namen nach Verlassen, Server Stop

Source Code

Beim Einkauf wurde der Code ausreichend kommentiert. Änderungen nach dem Code wurde mit Änderungsdatum und weiteren Kommentaren vermerkt.

Shell Commands zur Benutzung des Servers

Shell-Commands

- There are the following commands:

Run the server first

- Set an absolute or relative path to a viable `characters.json` file.

```
$ --config-charset <Path>
$ -c <Path>
```

- Set an absolute or relative path to a viable `.match` file.

```
$ --config-match <Path>
$ -m <Path>
```

- Set an absolute or relative path to a viable `.scenario` file.

```
$ --config-scenario <Path>
$ -s <Path>
```

- Sets the server capacity. Must be an integer between 3 and 500. Default is 20.

```
$ --x cap <int>
```

- Sets the time period for which the request of replays after a game is possible in minutes. Default is 5 minutes.

```
$ --x replay <int>
```

- Sets the maximum time for an NPC to take for a move in seconds. Default is 10 seconds.

```
$ --x npctime <int>
```

- Configures the log output to the console. Default is false.

```
$ --x clog <boolean>
```

- Configures the strictness, for the handling of possibly delayed messages(e.g. two players unpausing at the same time).

```
$ --x strict <boolean>
```

- Display the man/help list.

```
$ --help
```

```
$ -h
```

- Set the port the server will run on

```
$ --port <int>
```

```
$ -p <int>
```

- Must be followed up with an integer that will determine the detail of the written logs. Must be an integer between 0 and 6. Default value is 3 - INFO.

```
$ --verbosity <int>
```

```
$ -v <int>
```

Installation

→ Über die Kommandozeile: java -jar wordgame-0.0.1-SNAPSHOT-jar-with-dependencies.jar

→ wordgame-0.0.1-SNAPSHOT-jar-with-dependencies.jar befindet sich im Ordner wordgame2\target

→ Bemerkung: im .\target\default\ Ordner befinden sich die Konfigurationsdateien