



ulm university universität
uulm

**Fakultät für
Ingenieurwissenschaften,
Informatik und
Psychologie**

Institut für Software-
technik und Program-
miersprachen

Softwaregrundprojekt Meilenstein 3

Softwaregrundprojekt an der Universität Ulm

Vorgelegt von:

Gruppe 10

Dozent:

Florian Ege

Betreuer:

Stefanos Mytilineos

2019

Inhaltsverzeichnis

1	Schnittstellenarten, Dialoge und Dialogstruktur	3
1.1	Team- und Partiekonfigurator	3
1.1.1	Schnittstellenarten	3
1.1.2	Dialoge	3
1.1.3	Dialogstrukturdiagramme	4
1.2	Server	4
1.2.1	Schnittstellenarten	4
1.2.2	Dialoge	4
1.2.3	Dialogstrukturdiagramme	5
2	Grafische Gestaltung und Nutzungskonzept	5
2.1	Team- und Partiekonfigurator	5
2.1.1	Konfiguratormenü	5
2.1.2	Teammenü	6
2.1.3	Team bzw. Partiekonfiguration laden	6
2.1.4	Teamkonfigurator	7
2.1.5	Team bzw. Partiekonfiguration speichern	8
2.1.6	Konfiguration erfolgreich	8
2.1.7	Konfiguration ungültig	9
2.1.8	Partiemenü	9
2.1.9	Partiekonfigurator	10
2.2	Server	10
2.2.1	ffServerInit	10
2.2.2	ffServerRunning	11
2.2.3	ffServerError	11

1 Schnittstellenarten, Dialoge und Dialogstruktur

1.1 Team- und Partiekonfigurator

Diese Komponente enthält einen Konfigurator, mit dem sich sowohl Quidditch-Teams, als auch Partien konfigurieren lassen.

1.1.1 Schnittstellenarten

Als Benutzerschnittstelle wird, wie im Lastenheft vorgeschrieben, eine GUI verwendet.

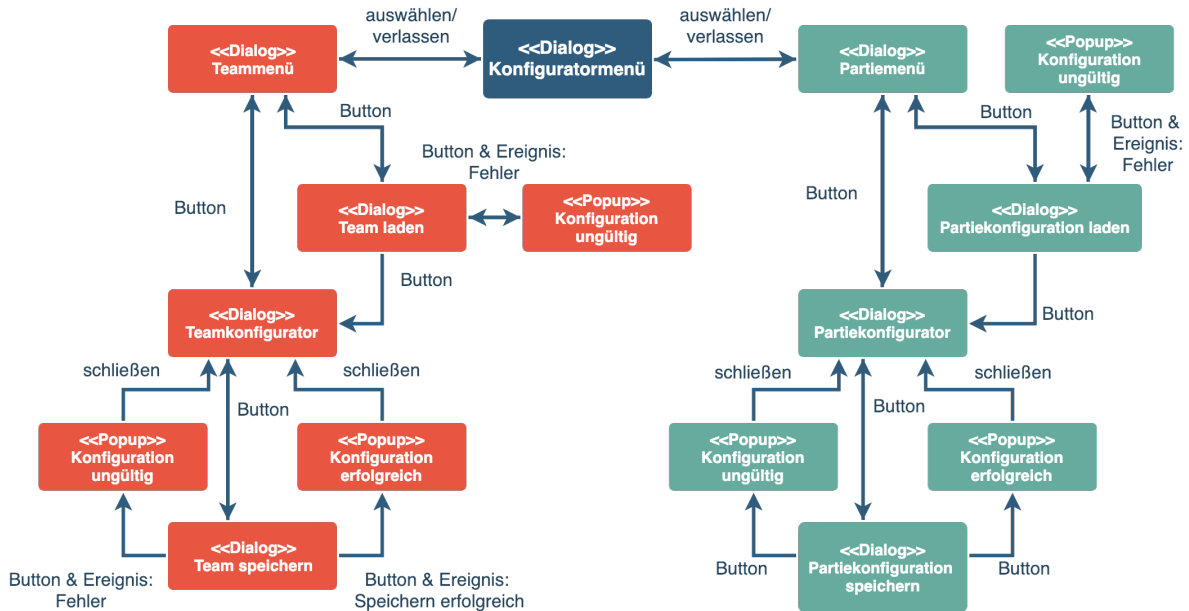
Begründung: Der Nutzer möchte alle Informationen zu einer gegebenen Konfiguration übersichtlich dargestellt bekommen und anhand dieser Darstellung direkt Änderungen vornehmen können. Hierfür ist eine GUI die intuitivste und sinnvollste Variante, da eine grafische Darstellung eine übersichtliche Visualisierung erlaubt und eine Änderung direkt anhand dieser Visualisierung möglich ist. Zudem könnte auf diese Weise die Komponente reibungslos in die Client-Anwendung integriert werden, falls man zu einem späteren Zeitpunkt den Konfigurator im Menü der Client-Anwendung direkt aufrufen möchte.

1.1.2 Dialoge

Im Folgenden werden den bereits formulierten Anforderungen und Anwendungsfällen der Komponente den entsprechenden Dialogen zugeordnet.

Name	Typ	Abgedeckte Anwendungsfälle	
Konfiguratormenü	Dialog	QA18	implizit aus Benutzerfreundlichkeit
Teammenü	Dialog	QA18	implizit aus Benutzerfreundlichkeit
Team laden	Dialog	FA72	Konfiguration öffnen
Teamkonfigurator	Dialog	FA70-72	Konfiguration erstellen/bearbeiten
Team speichern	Dialog	FA72	Konfiguration speichern
Partiemenü	Dialog	–	implizit, da Strukturierung erforderlich
Partiekonfiguration laden	Dialog	FA72	Konfiguration öffnen
Partiekonfigurator	Dialog	FA70-72	Konfiguration erstellen/bearbeiten
Partiefonfiguration speichern	Dialog	FA72	Konfiguration speichern
Konfiguration erfolgreich	Popup	QA17-18	Benutzerfreundlichkeit und Robustheit
Konfiguration ungültig	Popup	QA17-18	Benutzerfreundlichkeit und Robustheit

1.1.3 Dialogstrukturdiagramme



1.2 Server

Diese Komponente stellt den Server dar. Er hostet Spiele und enthält die Spiellogik. Zu Beginn wird er einmalig vom Systemadministrator gestartet und steht anschließend den Spielern von *Fantastic Feasts* zur Verfügung.

1.2.1 Schnittstellenarten

Als Benutzerschnittstelle wird ein CLI verwendet. **Begründung:** Der Nutzer kommt mit dieser Komponente über keine Benutzerschnittstelle in Berührung. Deswegen spielen Look and Feel prinzipiell keine Rolle. Die Funktionalität und Praktikabilität stehen im Mittelpunkt. Da der Systemadministrator in der Lage sein sollte, eine Konsole zu bedienen und dies unter Umständen auch vorzieht, besteht von dieser Seite keine Einschränkung. Zudem sind die Interaktionsmöglichkeiten mit dem Server begrenzt – es werden nur wenige Parameter benötigt – was zusätzlich für die Einfachheit einer CLI spricht.

1.2.2 Dialoge

Im Folgenden wird die Anforderung an den Server einem CLI-Dialog zugeordnet.

Name	Typ	Abgedeckte Anwendungsfälle	
ffServerInit	CLI Befehl mit Params	FA74	Serverkonfiguration einstellen
ffServerRunning	Response	FA74	Serverkonfiguration Feedback.
ffServerError	Response	FA74	Serverkonfiguration Feedback.

1.2.3 Dialogstrukturdiagramme

Da es sich um eine einfache Bedienung in der Konsole handelt, ist ein Dialogstrukturdiagramm nicht sinnvoll. Prinzipiell kann die Struktur aber wie folgt beschrieben werden: **ffServerInit**: Der Server lässt sich in der Konsole mit dem Namen der Server-Anwendung und zwei Parametern aufrufen. Der erste Parameter ist die Port-Nummer, über die der Server erreichbar ist. Im zweiten Parameter kann eine gültige Partiekonfigurationsdatei angegeben werden. Darauf gibt es zwei mögliche Antworten. **ffServerRunning**: War die Initialisierung erfolgreich antwortet der Server mit einer entsprechenden Nachricht in der u. a. die wichtigsten Parameter zusammengefasst werden. **ffServerError**: Im Falle eines Fehlers bei der Initialisierung wird mit einer Fehlermeldung geantwortet.

2 Grafische Gestaltung und Nutzungskonzept

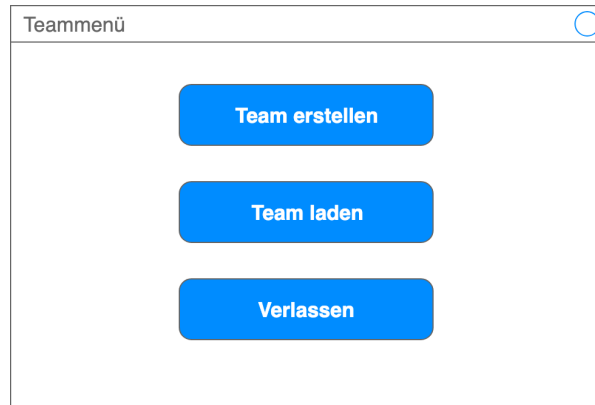
2.1 Team- und Partiekonfigurator

2.1.1 Konfiguratormenü



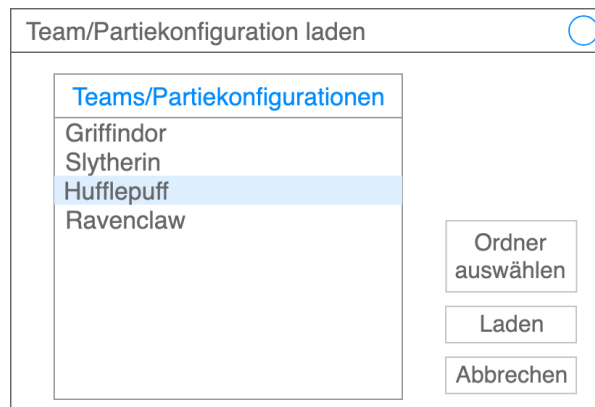
Über das Konfiguratormenü kann eine Auswahl zwischen dem Teammenü und dem Partiemenü über die entsprechenden Buttons getroffen werden. Diese führen zu den Menüs der jeweiligen Konfiguratoren. Über den Button *Verlassen* kann der Konfigurator verlassen werden.

2.1.2 Teammenü



Über das Teammenü kann eine Auswahl zwischen dem Laden und dem Erstellen einer Teamkonfiguration getroffen werden. Das erfolgt über die entsprechenden Buttons. Über den Button *Verlassen* kann das Teammenü verlassen werden.

2.1.3 Team bzw. Partiekonfiguration laden



Über diesen Dialog kann eine bereits vorhanden Konfigurationsdatei ausgewählt und im Konfigurator geladen werden. Das erfolgt über ein Dateiauswahlelement und die entsprechenden Buttons. Über den Button *Abbrechen* gelangt man zurück zu den entsprechenden Menüs. Da das Laden einer Teamkonfiguration nahezu identisch zum Laden einer Partiekonfiguration ist, wurde diese beiden Fälle in einem zusammen gefasst. Sollte eine zu ladende Konfigurationsdatei ungültig sein, öffnet sich das Popup *Konfiguration ungültig*.

2.1.4 Teamkonfigurator

Teamkonfigurator

Teamname

Teamfarbe

Teamfarbe (Ersatz)

Teamlogo

Fan

Fan

Fan

Fan

Fan

Fan

Fan

Fan

Fan

Fan

Fan

Speichern

Abbrechen

Jäger

Treiber

Hüter

Sucher

Name

männlich

weiblich

Besen

Name

männlich

weiblich

Besen

Name

männlich

weiblich

Besen

Name

männlich

weiblich

Besen

Name

männlich

weiblich

Besen

Name

männlich

weiblich

Besen

Name

männlich

weiblich

Besen

Im Teamkonfigurator können alle Parameter eines Teams eingestellt werden. Team- und Spielernamen lassen sich durch ein Textfeld bearbeiten. Teamfarben sind über eine Farbauswahl einstellbar. Das Teamlogo lässt sich aus einer Liste vorhandener Logos auswählen. Die Kästen mit den Kreuzen dienen als Platzhalter für Icons, die die verschiedenen Fans und Spieler darstellen und somit unterscheidbar machen. Fans sowie Besen der Spieler sind über eine Dropdown-Auswahl einstellbar. Das Geschlecht der Spieler lässt sich über Radio-Buttons einstellen. Bei jeder Änderung werden die entsprechenden Bedingungen für eine gültige Konfiguration geprüft und der Nutzer erhält visuelles Feedback (z. B. in Form von roter Schriftfarbe in den entsprechenden Feldern). Über den Button *Abbrechen* lässt sich der Konfigurator jederzeit verlassen. Ist eine gültige Auswahl eingestellt, kann die Konfiguration über *Speichern* in einem separaten Speicherdialog persistiert werden.

7

2.1.5 Team bzw. Partiekonfiguration speichern

The dialog box is titled 'Team/Partiekonfiguration speichern' and features a close button (blue circle) in the top right corner. It contains a list box on the left with the header 'Ordner' and two entries, 'Datei1' and 'Datei2'. To the right of the list box is a text input field containing 'Teamname.json'. Below the input field are three buttons: 'Ordner auswählen', 'Speichern', and 'Abbrechen'.

In diesem Dialog kann Dateiname und Speicherort der Konfiguration festgelegt werden. Der Button *Abbrechen* bringt den Nutzer direkt zurück in den entsprechen Konfigurator. Durch Klicken auf den Button *Speichern* wird die Datei mit dem gewählten Namen und Speicherort gespeichert.

Wurde versucht eine Konfiguration mit ungültigen Parametern zu speichern oder trat beim Speichervorgang ein Fehler auf, wird der Nutzer über dieses Popup darüber informiert. Der *Ok*-Button führt zurück zum entsprechenden Konfigurator.

2.1.6 Konfiguration erfolgreich

The dialog box is titled 'Konfiguration erfolgreich' and features a close button (blue circle) in the top right corner. The main text area contains the message 'Die Konfiguration war erfolgreich und wurde gespeichert.' Below the message is a single button labeled 'OK'.

Wenn alle Paramter einer Konfiguration gültig waren, wird dieser Dialog angezeigt. Der *Ok*-Button führt zurück zum entsprechenden Menü.

2.1.7 Konfiguration ungültig



Wurde versucht eine Konfiguration mit ungültigen Parametern zu speichern oder trat beim Speichervorgang ein Fehler auf, wird der Nutzer über dieses Popup darüber informiert. Der *Ok*-Button des Popups führt zurück zum entsprechenden Dialog.

2.1.8 Partimenü



Über das Partimenü kann eine Auswahl zwischen dem Laden und dem Erstellen einer Partiekonfiguration getroffen werden. Das erfolgt über die entsprechenden Buttons. Über den Button *Verlassen* kann das *Partimenü* verlassen werden.

2.1.9 Partiekonfigurator

Partiekonfigurator

Partikonfigname

Rundenanzahl bis Überlänge

100

Zeitspanne für Rundenaktionen

Aktion1

100 s

Aktion2

100 s

Ereigniswahrscheinlichkeiten

Ereignis1

0,15

Ereignis1

0,50

Ereignis1

0,33

Weitere Parameter können folgen...

Speichern

Abbrechen

Im Partiekonfigurator können alle Parameter für eine gültige Partiekonfigurationsdatei eingestellt werden. Dazu gehören unter anderem die Rundenanzahl, bis Überlänge erreicht ist, die Zeitspannen für die jeweiligen Spielaktionen und Ereigniswahrscheinlichkeiten. Je nach Art des Parameters sind Spinner, Slider, Textfelder oder im weiteren Verlauf der Implementierung noch andere Auswahllemente vorhanden.

Über den Button *Speichern* gelangt man in den *Partiekonfiguration speichern*-Dialog. Der Button *Abbrechen* führt zurück ins *Partiemenü*.

2.2 Server

2.2.1 ffServerInit

```
$ ffserver 1230 standard_partie.json  
... pending
```

Das Beispiel für einen Aufruf des Servers mit entsprechenden Parametern dar.

2.2.2 ffServerRunning

Fantastic Feasts server is running on Port 1230 ...

Im Falle einer erfolgreichen Initialisierung wird dies dem Systemadministrator über diese Nachricht mitgeteilt. In dieser Nachricht können weitere Informationen über den initialisierten Server verpackt werden.

2.2.3 ffServerError

Error: Initialization failed

Im Falle eines Fehlers bei der Initialisierung wird der Systemadministrator mit dieser Fehlermeldung benachrichtigt.