

# **Dokumentation**

## *Galgenmännchen*

### *Torben Spiekermann*

## **1 Teammitglieder**

Torben Spiekermann, (Ursprünglich: Jonas Wrede, Eric Tika)

## **2 Einleitung**

Galgenmännchen ist ein Spiel für mindestens 2 Spieler. Das Spiel kann in einem lokalen Netzwerk gespielt werden. Der Spielmacher legt als erstes ein Wort fest, das von den anderen Spielern erraten werden muss. Dafür haben sie 12 Fehlversuche frei. Gewonnen hat derjenige, der entweder das Lösungswort oder den letzten fehlenden Buchstaben schreibt. Der Gewinner darf sich beim automatisch startenden nächsten Spiel das Wort aussuchen. Wird das Wort nicht gefunden (Leben werden auf 0 reduziert) startet ein neues Spiel mit gleichbleibender Rollenverteilung.

## **3 Produktbeschreibung**

Ziel des Spiels ist es, vor allen Mitspielern als erstes das Lösungswort zu erraten, indem man Buchstaben oder direkt das vom Spielmacher festgelegte Lösungswort an den Server sendet, welcher die Eingabe verarbeitet und ein Feedback gibt.

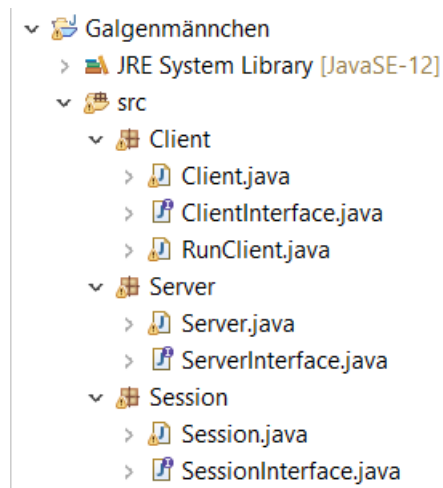
### **3.1 Funktionsumfang**

Zu Beginn des Programms ist die Eingabe des Nutzernamens über ein JOption Dialog Fenster erforderlich. Daraufhin wird eine Spielsitzung gestartet, die als Oberfläche ein Eingabefeld, ein Textfeld zur Anzeige aller „Nachrichten“ sowie nach Eingabe des Spielers das Lösungswort verschlüsselt und die übrigen Leben anzeigt.

Bei diesem Spiel können beliebig viele Spieler gegeneinander spielen, wichtig ist, dass es mindestens 2 Spieler sind. Die Spieler können sowohl einzelne Buchstaben sowie ganze Wörter als Lösungsvorschläge senden. Auf Serverseite wird geprüft, ob die Lösungsvorschläge richtig oder falsch sind. Wenn sie falsch sind werden Leben abgezogen. Wird ein richtiger, also ein im Lösungswort vorhandener Buchstabe gesendet, werden die Buchstaben unten im Lösungswort an entsprechender Stelle angezeigt. Grundsätzlich sehen alle Spieler die Eingaben aller anderen Spieler (außer natürlich die Eingabe des Lösungswortes). Wird das Lösungswort direkt durch Eingabe der richtigen Lösung oder indirekt durch Eingabe des letzten fehlenden Buchstaben gefunden startet der Server automatisch das nächste Spiel, der Gewinner des vorherigen Spiels darf nun das neue Lösungswort festlegen. Wird die Lösung vor Ablauf aller 12 Leben nicht gefunden, startet ein neues Spiel mit gewohnten Rollen.

### 3.2 Aufbau der Software

Die Software ist auf 3 Packages aufgeteilt: Server, Client und Session. Im Client Package befinden sich 2 Klassen (RunClient und Client) und ein Interface. Im Server Package befindet sich ein Interface (ServerInterface) und eine Klasse (Server). Ebenso befinden sich im Session Package ein Interface (SessionInterface) und eine Klasse (Session). Zu beachten ist, dass alle Interfaces von Remote erben. Außerdem erben alle Klassen, die die Interfaces implementieren, von UnicastRemoteObject. Somit existieren in dieser Anwendung 3 Typen von Server Objekten. Dies ermöglicht die direkte Kommunikation zwischen allen oder einzelnen Mitspielern.



#### Client Package

In diesem Package befinden sich alle für die Nutzer notwendigen Methoden. Dazu gehört das Laden der Oberfläche sowie die Methoden, auf die der Server zugreifen können muss.

#### ClientInterface

Das ClientInterface dient der Server-Client Kommunikation und definiert die Methoden, die auf Serverseite ausgeführt werden um Aktionen auf Clientseite hervorzurufen. Das ClientInterface definiert 4 Methoden:

1. getMessage(String name, String message)
2. newGame(boolean gamemaker)
3. getGamemaker()
4. showTmpSolution(String solution)

#### Client

Die Client Klasse implementiert das ClientInterface. In dieser Klasse wird die ganze Zeit mit einem RunClient Objekt gearbeitet, welches den Nutzer darstellt. So gelingt die Überbückung von Aufruf auf Serverseite zu Ausführung bei jedem einzelnen Client.

#### RunClient

Diese Klasse ist die Klasse, die der Nutzer ausführt wenn er das Spiel starten möchte. Nach Eingabe des Namens durch ein JOptionPane wird die Spieloberfläche geleaden. In dieser Klasse wird definiert, was nach Eingaben auf der Benutzeroberfläche geschehen soll.

### Server Package

In diesem Package befindet sich der „Hauptserver“ und das Interface dazu. Da die Client-Server Kommunikation hauptsächlich über die einzelnen Sitzungen (Session) geschieht, besitzt das ServerInterface nur eine Methode.

### ServerInterface

Das ServerInterface besitzt nur die Methode `makeSession(String name, ClientInt clientInt)`, welche als Rückgabewert ein `SessionInterface` hat.

### Server

Die Server Klasse implementiert das `ServerInterface` und muss als erstes ausgeführt werden, weil dies der Server ist mit dem sich die `RunClients` verbinden. Der Server verarbeitet die Eingaben der einzelnen Nutzer. So wird hier das Lösungswort gespeichert, die Eingaben der Nutzer mit dem Lösungswort verglichen, Leben reduziert etc.

### Session Package

Die Kommunikation von Client zu Server geschieht über Sessions. Hier werden die Methoden deklariert, die vom Client aufgerufen aber auf Serverseite verarbeitet werden.

### SessionInterface

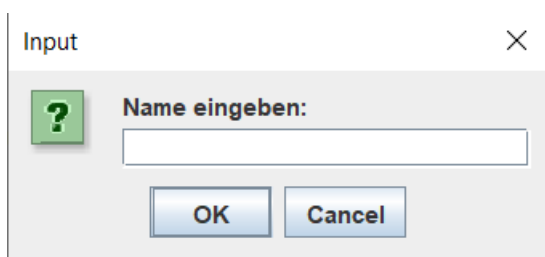
Dieses Interface besitzt 2 Methoden: `sendMessage(String message, boolean gamemaker)` und `getId()`. Wichtig ist vorallem die `sendMessage` Methode, weil der Client auf diese zugreift um dem Server eine Nachricht zu senden

### Session

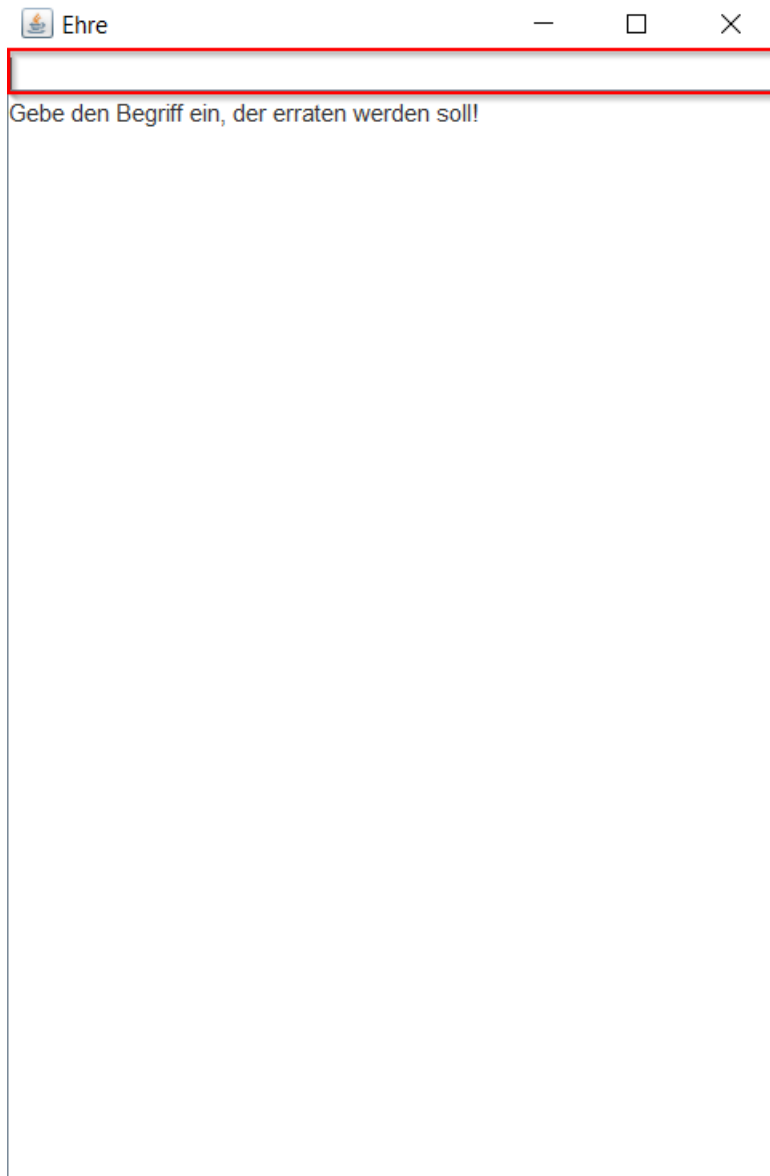
Die Session Klasse implementiert das `SessionInterface`. Sie stellt Methoden bereit, um Sessions zu vergleichen und um Nachrichten an den Server zu senden. Je nachdem welche Rolle (gamemaker) der Client hat, wird entweder die `sendSolution()` oder `sendAnswer()` Methode auf dem Server ausgeführt

## 4 Bedienungsanleitung

Um das Spiel zu spielen muss als erstes die Server Klasse gestartet werden. Anschließend können beliebig viele Nutzer durch die Ausführung der `RunClient` Klasse dem Server und somit dem Spiel beitreten. Nach Starten der `RunClient` Klasse muss der Nutzernamen eingegeben werden:



Der Spieler, der als erstes dem Server beitrifft, ist als erstes der „Spielmacher“ und darf die erste Lösung festlegen. Dies geschieht, indem er in das Textfeld oben (unten rot) das zu erratene Wort eingibt und mit Enter bestätigt. Achtung: Leerzeichen dürfen nicht im Wort enthalten sein. Die Groß/Klein-schreibung ist egal, da der Server alle Buchstaben zu Großbuchstaben macht.



The image shows a window titled "Ehre" with standard Windows window controls (minimize, maximize, close). Below the title bar is a text input field with a red border. Below the input field is a large, empty rectangular area. The text "Gebe den Begriff ein, der erraten werden soll!" is visible at the top of this large area.

Bis zur Eingabe des Lösungswort können alle anderen Spieler miteinander chatten. Wenn der Spielmacher das Lösungswort eingegeben hat bekommen alle Mitspieler eine Benachrichtigung im Chatfenster (unten rot).



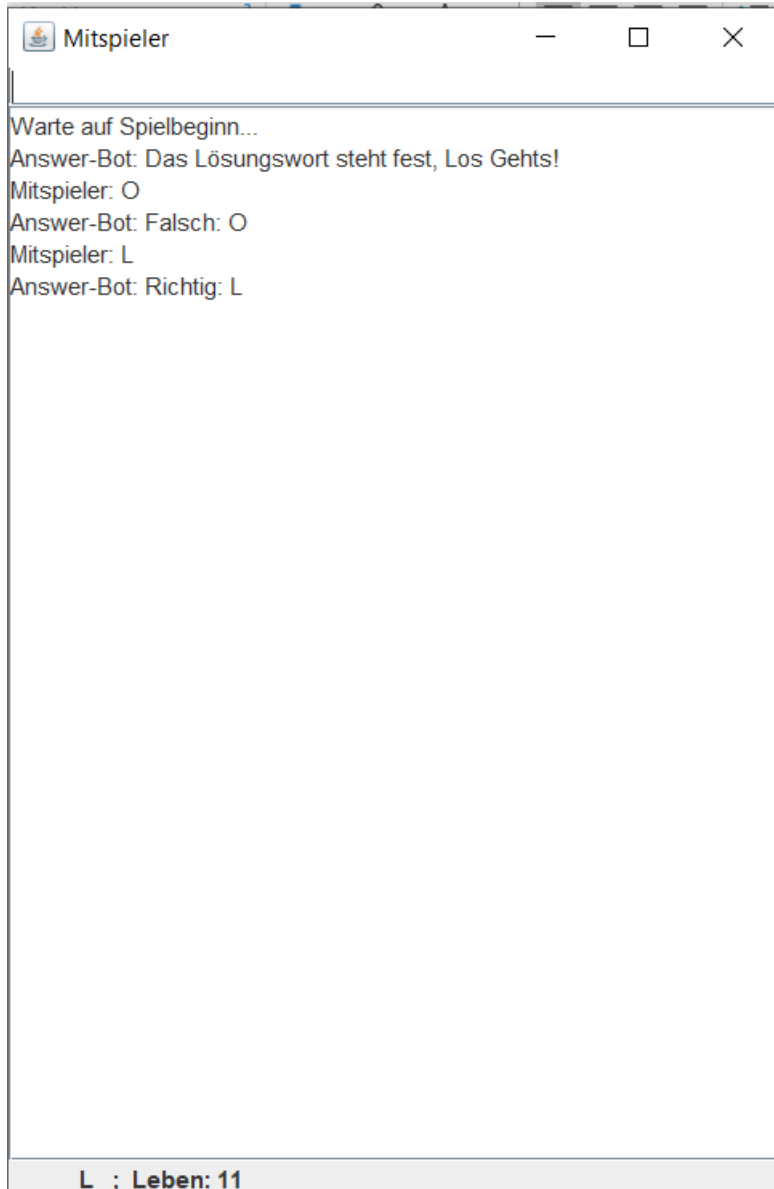
Nach Eingabe des Lösungsworts wird am unteren Fensterrand das Lösungswort „verschlüsselt“ und die übrigen Leben angezeigt. Nun können alle Mitspieler durch Eingabe einzelner Buchstaben oder mit Lösungswortvorschlägen versuchen, die richtige Lösung zu erraten. In diesem Beispiel ist das Lösungswort Esel und es spielen 2 Spieler. Bei allen Spielteilnehmern sieht das Chat Fenster gleich aus.

Beispiel falsche Eingabe:



Nach der falschen Eingabe (O ist nicht in Esel) meldet sich der Answer-Bot und sagt das O Falsch ist.

Beispiel richtige Eingabe:



Nach der richtigen Eingabe meldet sich der Answer-Bot und sagt das L richtig ist. Wenn das ganze Lösungswort unten gefüllt ist oder jemand die richtige Lösung eingibt startet ein neues Spiel.

## 5 Aufgabenverteilung

Da ich das einzige Mitglied dieser Gruppe bin stammt das gesamte Projekt von mir. Mir ist bewusst, dass die Oberfläche schöner hätte sein können. Ich war auch gerade noch an der Methode, die das Galgenmännchen zeichnet, jedoch habe ich dies aus Zeitgründen nicht mehr geschafft.