TypefastClient Documentation

Overview

`TypefastClient` is a Java Swing-based graphical client for connecting to the Typefast server. The application allows users to register, login, join a game, view scores, and logout. The client communicates with the server using socket programming.

Main Components

- **GUI Components**: JFrame, JTextArea, JTextField, JButton, JDialog
- **Networking Components**: Socket, BufferedReader, PrintWriter

Constants

- **SERVER_ADDRESS**: The address of the server
- **SERVER_PORT**: The port number of the server (default is 12345).
- **Command Strings**: Commands sent to the server (`**LOGIN_CMD**`, `**REGISTER_CMD**`, `**JOIN_CMD**`, `**LOGOUT_CMD**`, `**SCOREBOARD_CMD**`, `**EXIT_CMD**`).

Fields

- `**Socket socket**`: The socket used to communicate with the server.
- `**BufferedReader in**`: Reader for incoming messages from the server.
- `**PrintWriter out**`: Writer for sending messages to the server.
- **GUI Components**: `**JFrame frame**`, `**JTextArea messageArea**`, `**JTextField userInputField**`, `**JButton loginButton**`, `**JButton registerButton**`, `**JButton joinButton**`, `**JButton logoutButton**`, `**JButton scoreboardButton**`, `**JButton exitButton**`.

Methods

Main Method

```java
public static void main(String[] args) {
```

- Entry point of the application. Initializes and shows the GUI on the Event Dispatch Thread (EDT).

Constructor

```java
public TypefastClient() throws IOException {  1 usage
```

- Establishes a connection to the server. If the connection fails, it displays an error message.

GUI Initialization

```java
private void createAndShowGUI() {  1 usage
```

- Sets up the main frame and its components.
- Creates input and button panels.
- Configures buttons and their action listeners.
- Starts a thread to listen for messages from the server.

Server Message Handling

```
private void handleServerMessage(String message) {  1 usage
```

- Processes messages from the server and updates the GUI accordingly.
- Enables/disables buttons based on the message content.
- Handles specific commands like "Login successful", "Game over", "Logout successful", "Added to waiting list", "Exiting game", "Score for this word", and "Scoreboard".

User Input Handling

- The `userInputField` sends user input to the server when the Enter key is pressed.

Button Actions

- Login Button: Shows the login form.
- Register Button: Shows the registration form.
- Join Button: Sends a join request to the server.
- Logout Button: Sends a logout request to the server.
- Scoreboard Button: Requests the scoreboard from the server.
- Exit Button: Sends an exit request to the server.

Popup Messages

Scoreboard Popup

```
private void showScoreboardPopup(String formattedScoreboard) {  1 usage
```

- Displays the scoreboard in a popup dialog.

Login Form

```
private void showLoginForm() {   1 usage
```

- Displays a modal dialog for user login.
- Sends login credentials to the server.

Register Form

```
private void showRegisterForm() {   1 usage
```

- Displays a modal dialog for user registration.
- Sends registration details to the server.

Example Usage

1. **Run the Client**: Execute the `TypefastClient` class. The GUI will appear.

2. **Login**: Click the "Login" button and enter your credentials.

3. **Register**: Click the "Register" button if you need to create a new account.

4. **Join Game**: Once logged in, click the "Join" button to enter a game.

5. **View Scoreboard**: Click the "Scoreboard" button to view scores.

6. **Logout**: Click the "Logout" button to log out.

7. **Exit**: Click the "Exit" button to leave the game and return to the dashboard.

Error Handling

- Connection Errors: Displays a dialog if the connection to the server cannot be established.
- Input Validation: Ensures that username and password fields are not empty during login and registration.

Conclusion

`TypefastClient` provides a user-friendly interface for interacting with the Typefast server. It leverages Java Swing for the GUI and socket programming for network communication. This client allows users to perform essential actions such as logging in, registering, joining games, viewing scores, and logging out efficiently.