Grobe Lösungsansätze für GUI aus JavaFX-Komponenten

**1. Aufbau der GUI aus JavaFX-Komponenten**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **GUI** **Aufbau** | **Eingabefeld** | **Buttons** | **Anmerkung** | **Layout** |
| *Initial Prompt* | ja | Submit | * Farbe der Buttons richtig auswählen | “VBox” oder “GridPane” |
| *Main Menu* | nein | start game  profile setting  stats  exit | * Farbe der Buttons richtig auswählen |
| *Game Window* | ja | Profile Setting  Stats  exit | * Braucht Textanzeige * zeigt #WPM, #Errors, und Top players |
| *Result Window* | nein | play again  exit | * “TableView” für die Anzeige der Top-Spieler, #WPM, und #Errors |
| *Player Stats Window* | nein | exit  reset stats | * “TableView” für die Anzeige der Spielrunde, durchnitliche WPM, #Errors, beste WPM, und durchnitliche Genauigkeit (Prozent) |
| *Profile Setting Window* | ja (3) | save  cancel | * Farbe der Buttons richtig auswählen |

**2. Übertragung des Spielzustands**

A diagram of a game client

Description automatically generated

- Wie wird der Spielzustand übertragen?

- Netzwerkprotokoll: Verwenden von JSON über HTTP oder WebSocket für die Kommunikation zwischen Client und Server.

- Serialisierung: Spielzustandsdaten werden in JSON-Format serialisiert und über das Netzwerk gesendet.

- Deserialisierung: Empfangene JSON-Daten werden deserialisiert und in Java-Objekte umgewandelt.

**3. Implementierung der GUI Mockups**

***- Schritt 1: Strukturplanung***

- Layout-Auswahl: Bestimmen, welche Layouts (z.B. “VBox”, “HBox”, “GridPane”) für welche GUI-Komponenten am besten geeignet sind.

- Komponentenauswahl: Auswählen der benötigten JavaFX-Komponenten (z.B. “TextField”, “Button”, “Label”).

***- Schritt 2: Komponentenplatzierung***

- Mockup-Umsetzung: Platzieren der Komponenten gemäß den Mockups in den gewählten Layouts.

- Event-Handler: Implementieren von Event-Handlern für Benutzeraktionen (z.B. Knopf-Klicks).

***- Schritt 3: Datenbindung und Validierung***

- Datenbindung: Verwenden von JavaFX-Properties und Bindings für die Echtzeit-Aktualisierung der GUI basierend auf dem Spielzustand.

- Validierung: Implementieren von Eingabevalidierungen für die Benutzereingabe (z.B. Benutzernamen-Validierung).

- ***Schritt 4: Integration und Testen***

- Integration: Verbinden der GUI-Komponenten mit den Backend-Services für die Spielzustandsübertragung.

- Testen: Durchführen von Funktionstests und Benutzerakzeptanztests, um sicherzustellen, dass die GUI wie erwartet funktioniert.