Studienleistung 2

Alexander Bazo und Jakob Fehle

You vs. Will

In dieser Aufgabe entwickeln Sie ein Spiel, mit dem Nutzer und Nutzerinnen ihr Wissen über den Wortschatz Shakespeare testen können. Die SpielerInnen haben dabei eine Minute Zeit möglichst viele der Worte einzugeben, die in den Dramen William Shakespeares verwendet wurden. Das Spiel beginnt, sobald die SpielerInnen ein erstes Wort in das Input-Feld eingeben und die Enter-Taste betätigt wird. Das Spiel überprüft nun, ob das eingegebene Wort im Wortschatz vorhanden ist: Falls ja, wird das Wort als Wort-Treffer aufgelistet, andernfalls passiert nichts und das Eingabefeld wird geleert. Nach Ablauf von 60 Sekunden sind keine weiteren Eingaben mehr möglich und die SpielerInnen bekommen einen Ergebnistext angezeigt. Hierbei werden Informationen bezüglich der absoluten und prozentualen Wort-Treffer aufgelistet.

Vorgaben

Im Starterpaket finden Sie eine nahezu vollständiges HTML-Dokument. Eine passende CSS-Datei für die Aufgabe ist vorhanden und mit dem HTML-Dokument verknüpft. Ebenfalls vorhanden ist eine JavaScript-Datei (resources/js/index.js), die in der HTML-Datei aufgerufen wird. Beim Start der Anwendung wird automatisch eine Liste der relevanten Wörter aus einer JSON-formatierten Datei geladen und als JavaScript-Array bereitgestellt. Sobald dieser Prozess abgeschlossen ist, wird die Methode onWordlistAvailable aufgerufen. Hier können Sie mit der Arbeit am Quellcode beginnen. Verteilen Sie Ihren Javascript-Code ggf. auf mehrere Dateien (oder Module). Änderungen an der HTML-Struktur und den vorgegebenen CSS-Regeln sind nicht notwendig.

Starten der Anwendung

Um die Anwendung korrekt auszuführen, wird ein lokaler Webserver benötigt. Dieser sorgt beim Laden der Webseite im Browser dafür, das die Wortliste korrekt eingelesen wird und erlaubt bei Bedarf die Verwendung von ES6-Modulen für die Strukturierung des Codes. Dazu können Sie z.B. die Live Server-Extension für Visual Studio Code verwenden, die wir Ihnen bereits im Kurs vorgestellt haben.

Verpflichtende Anforderungen

- NutzerInnen könnne Wörter über das vorbereitete Eingabefeld eingeben. Die Eingabe eines Wortes wird dabei mit der Eingabetaste (*Enter*) abgeschlossen. Auf Eingabefeldern (`input`) können Sie einen change-Event registrieren, um über den Zeitpunkt informiert zu werden, an dem die NutzerInnen die Eingabe eines Worts mittels **Eingabetaste** abschließen.

- Für jedes eingegebene Wort wird geprüft, ob dies im eingelesenen Wortschatz vorhanden ist. Falls ja, wird ein neuer Eintrag in der HTML-Liste mit der CSS-Klasse `result-list` erstellt. Verwenden Sie für diese Einträge ein `<li>`-Element, das so aufgebaut ist: `<li><span class="count">HÄUFIGKEIT</span><span class="word">WORT</span></li>`. Im Element wird das Wort selber sowie dessen absolute Häufigkeit im Korpus angezeigt. Diese Information finden Sie in der Eingangs eingelesenen Wortliste. Wörter sollen nur einmal in der Ergebnissliste eingetragen werden. Ersetzen Sie hierfür den obigen Beispiel-String dynamisch mit realen Daten, indem Sie den Inhalt der beiden `<span>`-Elemente (*HÄUFIGKEIT* und *WORT*) austauschen.
- Schränken Sie die Zeit, in der NutzerInnene eingaben tätigen können auf 60 Sekunden ein. Dazu müssen Sie nach der Eingabe des ersten Wortes einen *Timer* starten, der Ihnen nach 60 Sekunden die Möglichkeit gibt, das Spiel zu beenden. Idealerweise implementieren Sie dazu den Start- und Endpunkt Ihres Spiels bzw. der Raterunde in zwei separaten Methoden (z.B. `startRound` und `endRound`). Die Funktion 'setInterval()' erlaubt Ihnen eine Methode innerhalb Ihres Codes anzugeben, die nach einer bestimmten Zeit aufgerufen wird. Nutzen Sie diese beim Starten der Raterunde, um nach den 60 Sekunden das Spielende verarbeiten zu können.
- Nach Ablauf der 60 Sekunden sollen keine weiteren Eingaben mehr möglich sein. Im HTML-Element mit der Klasse score wird den SpielerInnen ein kurzer Informationstext angezeigt, der mitteilt, wie viele Wörter sie erraten haben. Geben Sie das Ergebnis sowohl absolut als auch prozentual (auf die Gesamtanzahl der Wörter bezogen) an.
- Achten Sie auf eine hohe Codequalität. Verwenden Sie passende und verständliche Bezeichner. Kommentieren Sie Ihren Code dort wo nötig. Trennen Sie die verschiedenen Aufgabenbereiche Ihrer Anwendung voneinander und nutzen Sie dazu z.B. unterschiedliche JavaScript-Dateien.

Optionale Anforderungen

- Während der Spielrunde wird die noch verbleibende Zeit im Format MM:SS im HTML-Element mit der Klasse `time-output` angezeigt. Sie können hierfür erneut die `setInterval()`-Funktion verwenden. Exakte Daten, Zeitpunkte und Zeitabstände können Sie mit Hilfe des Objekts `Date` ermitteln. Nehmen Sie hierfür eine geeignete Einheit, um die vergangene Zeit zu messen.
- Vereinfachen Sie die Anlegung eines neuen Wortes in der HTML-Liste mit der CSS-Klasse `result-list`, indem sie die JavaScript-Funktionalität von Template Strings verwenden, um ein neues und bereits befülltes Element zu erzeugen, welches an die Liste angehängt werden kann.

Abgabekriterien:

Laden Sie Ihre Lösung bis spätestens 18.6.2021 (23:59 Uhr) als zip-komprimierten Ordner auf GRIPS hoch. Benennen Sie die einzelnen Dateien pro Aufgabe sinnvoll und verwenden Sie geeignete Formate:

- Aufgabe 1: Das gesamte Projekt (HTML, JS, CSS)

Der Name der Datei ergibt sich aus dem Präfix „SL_WT_SS21“, der Nr. der Studienleistung, ihrem Vor- und Nachnamen jeweils getrennt durch _ .

Beispiel: **SL_WT_SS21_2_Max_Mustermann.zip**