WBE-Praktikum 13

UI-Bibliothek (Teil 3)

Aufgabe 1: «Vier gewinnt» Abschluss (Mini-Projekt)

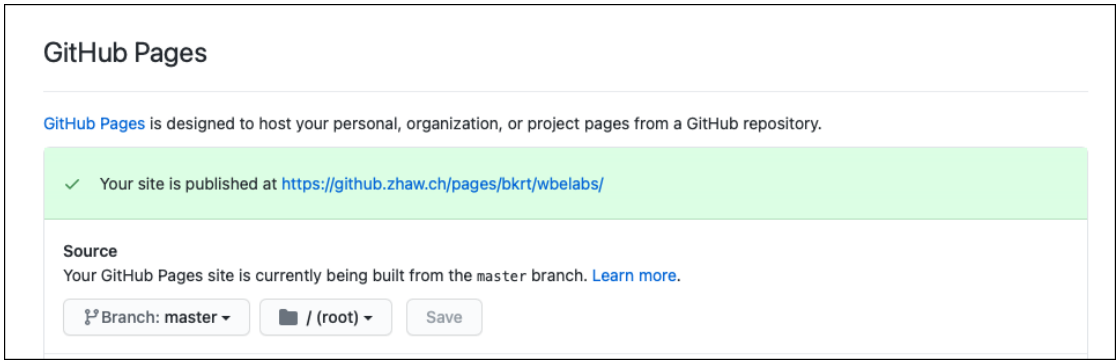
In dieser Aufgabe geht es darum, das Miniprojekt abzuschliessen. Es sollte nun spielbar sein und über eine Reihe von Funktionen verfügen wie:

- Möglichkeit zum Persistieren des Spielzustands
- Mehrfaches Undo
- Erkennen, wenn ein Spieler gewonnen hat

Sie können nun das Projekt noch nach Belieben verbessern oder verschönern (CSS…), bevor Sie es auf GitHub Pages zur Verfügung stellen.

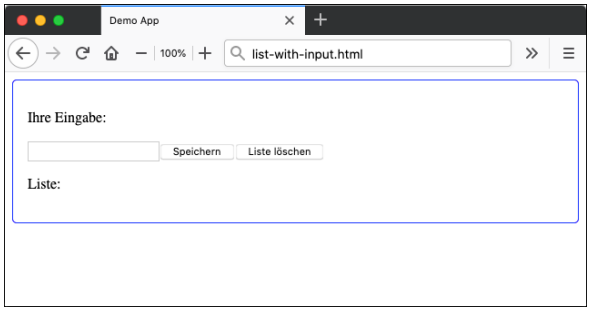
Hier die Aufgaben zum Abschluss des Projekts:

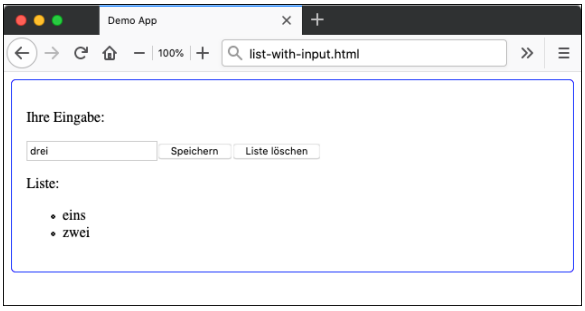
- Fakultativ (s. Aufgabe 3): verwenden Sie einen State-Hook zum Verwalten des Zustands.
- Stellen Sie Ihr Spiel unter GitHub Pages zur Verfügung: Da Sie Ihr Projekt sicher schon unter unter https://github.zhaw.ch verwalten, brauchen Sie in den Einstellungen nur noch GitHub Pages zu aktivieren (s.u.).
- Erstellen Sie eine kurze Doku (PDF, HTML, oder Markdown) mit Titel, wer zum Team gehört, einer Funktionsübersicht und was sonst für Ihre Implementierung erwähnt werden sollte. Auch ein oder mehrere Screenshots können nicht schaden. Insgesamt sollte die Doku aber nicht viel mehr als eine Seite umfassen. Ergänzen Sie die Seite mit dem Spiel um einen Link zur Doku.
- Stellen Sie den Link zu Ihrem Spiel Ihrem Betreuer im Praktikum zur Verfügung. Details dazu sind klassenspezifisch geregelt und werden in der Praktikumslektion besprochen.



Aufgabe 2: Liste aufbauen

Verwenden Sie die experimentelle Bibliothek SuiWeb zum Aufbau einer einfachen Seite: Diese besteht aus etwas Text, einem Eingabefeld, zwei Buttons und einer Liste, welche zunächst leer ist. Dann können einzelne Listenelemente über das Eingabefeld der Liste hinzugefügt werden. Ein Klick auf «Speichern» soll den Wert im Eingabefeld in die Liste übernehmen und das Eingabefeld löschen:





Hinweise

- Orientieren Sie sich an den Demos zum Unterricht.
- Das Umsetzen von „controlled input elements" sollte sowohl mit Version 1.1 als auch mit der älteren Version 0.3.4 von SuiWeb funktionieren.[1] Im Verzeichnis *lib* sind daher verschiedene SuiWeb-Versionen abgelegt.
- Implementieren Sie zwei Komponenten: eine Komponente *App* bildet den Rahmen der Applikation und eine Komponente *List* gibt die Liste aus.
- Verwenden Sie „State Hooks" in der *App*-Komponente, um die Zustände der Applikation zu speichern und zu aktualisieren.
- Der Button «Liste löschen» soll genau dies machen.
- Verschönern Sie die Darstellung (die Seite darf durchaus besser aussehen als die beiden Bilder oben). Dazu gibt es zwei Möglichkeiten. Selbstverständlich kann die Darstellung in einer separaten CSS-Datei beschrieben werden. Alternativ kann die Darstellung auch in JavaScript der Komponente hinzugefügt werden. Probieren Sie einmal die zweite Variante.
- Ergänzen Sie die Eingabe testweise so, dass nur Zahlen eingegeben werden können. Mit einem „controlled input element" sollte das kein grosses Problem sein.

[1] Grund: In Version 1.0 hat sich ein Bug beim Re-Rendern von *input*-Elementen eingeschlichen, der in dieser Implementierung schwierig zu beheben ist. Grund: beim Re-Rendern von interaktiven Elementen soll der letzte Zustand (Inhalt, Cursor-Position, Markierung) wieder abgebildet werden. Falls Si aus diesem Grund die ältere Version 0.3.4 verwenden, ist folgendes zu beachten: Die Funktion *useState* hat 3 Parameter: die ersten beiden dienen zur Identifikation des Zustands, der dritte ist der Initialwert. Ausserdem erhalten die *setState* Funktionen eine Update-Funktion als Argument anstelle des neuen Werts, z.B. setText(() => e.target.value) statt setText(e.target.value).

Aufgabe 3: «Vier gewinnt» mit *setState* (fakultativ, Mini-Projekt)

Fakultative Zusatzaufgabe

In der letzten Version wurde das Spielfeld von «Vier gewinnt» mit Hilfe von Komponenten generiert. Der Spielzustand wurde aber in einer globalen Variablen *state* gespeichert.

Nun soll der Zustand des Spiels sowie die Event Handler in der Container-Komponente *App* verwaltet werden. Dafür sind einige Umbauarbeiten nötig, speziell was die Ereignisbehandlung angeht.