**Aufgabe Sequenzmemory**

**Aufgabe:**

Konzipiere ein kleines Memoryspiel als Web-App für Desktoprechner, Tablets und Smartphones, bei dem sich ein Spieler die zufällige Position von Zeichen merken und diese in einer bestimmten Reihenfolge aufdecken muss, nachdem sie verdeckt wurden.

Zu Beginn eines Spiels kann der Spieler selbst eine Sequenz eingeben oder wählen, dass er mit einer von 10 (oder mehr) vordefinierten Sequenzen spielen will. Eine solche Sequenz könnte beispielsweise die Zeichenkette "EIA2-Inverted" sein. Weiterhin kann er die Zeit angeben, die ihm für das Spiel zur Verfügung steht und die Zeit, die ihm die zufällig durchmischte Zeichenfolge zu Beginn gezeigt wird. Im Beispiel könnte diese durchmischte Folge derart aussehen: "vAntIde-2IEre".

Dann startet das Spiel und die Zeichenfolge wird für die Dauer der angegebenen Zeit gezeigt, eine Interaktion ist in dieser Zeit nicht möglich. Dann werden die Zeichen, an gleicher Position verharrend, verdeckt. An ihrer Stelle sind nun gleichfarbige und gleichförmige Flächen zu sehen, die per Klick oder Touch aufgedeckt werden können. Aufgabe des Spielers ist es nun, die Zeichen in der korrekten Reihenfolge der Ausgangssequenz aufzudecken. Im Beispiel muss er also zuerst die Fläche an der Position des "E" aufdecken, dann eines der großen "I"s, dann das "A" usw. Verlässt der Spieler die korrekte Sequenz indem er ein falsches Zeichen aufdeckt, erhält er für zwei Sekunden ein sinnvolles optisches Feedback, danach werden alle Flächen wieder verdeckt. Innerhalb der vorgegebenen Zeit muss er die Sequenz vollständig aufdecken, sonst gilt das Spiel als verloren. Gelingt es ihm aber, erhält er positives optisches Feedback. Während des Spiels werden die Zeiten, rückwärts laufend, in Sekunden auf dem Bildschirm angezeigt.

Für den Spieler wird es voraussichtlich hilfreich und motivierend sein, wenn er zusätzlich permanent die korrekte Sequenz auf dem Bildschirm sieht und farblich markiert darin die Zeichen, die er gerade bereits erfolgreich aufgedeckt hat. Außerdem solle er noch eine kleine Hilfefunktion nutzen können: Genau einmal im Spiel kann er eine Taste auf dem Keyboard halten. Solange er diese hält, wenigstens aber zwei Sekunden, erscheinen die entsprechenden Zeichen in der durchmischten Sequenz aufgedeckt.

**Konzeption:**

Halte dich strikt an die in der Lektion geübte Reihenfolge und Syntax der Konzeption, also

1. Anwendungsfalldiagramm, welches einen Überblick über alle Interaktionen mit dem User und die wesentlichsten Vorgänge innerhalb des Systems darstellt,

2. UI-Skizze mit Angaben zu HTML-Tags, CSS-Selektoren, Signalen und Listeners,

3. Aktivitätsdiagramme, mit Datenstrukturen, den wesentlichsten Algorithmen, Listenerregistrierungen, Signalverarbeitungen und Subaktivitäten.

Achte sehr darauf, dass Du nicht lediglich die Aufgabenstellung in anderer Form wiederholst, sondern dass Du eine klare und mit der bislang in dieser Modulreihe geübten Technik durchführbare Lösung konzipierst. Eine andere mit der Technik vertraute Person muss anhand deiner Konzeption das Programm kodieren können, ohne über viel mehr als die reine Umsetzung in Code nachzudenken.

**Produktion:**

Suche dir eine Person im Kurs, mit der Du Konzepte tauschst. Jeder von euch versucht nun das Konzept der anderen Person in Code zu übersetzen. Bleibt miteinander in regem Kontakt um euch Rückmeldung zu der Qualität des Konzepts geben zu können. Schlüpft dabei in die Rolle eines Programmierers, der die Aufgabenstellung nicht kennt und daran auch nicht interessiert ist, sondern eine gute Konzeption erwartet, die er einfach "heruntercoden" kann. Das bedeutet, dass ihr Lücken im Konzept nicht durch eigene Lösungsfindung stopft, sondern den Konzepter um entsprechende Ergänzung und Korrektur bittet. Diese muss dann tatsächlich im Konzept umgesetzt werden, nicht nur mündlich überliefert. Am Ende sollen alle Datenstrukturen und Algorithmen auch im Konzept ausreichend weit herunter gebrochen sein.

**Recherche:**

Es könnte sehr hilfreich sein, wenn Du folgende Methoden und Attribute einiger Javascript-Objekte näher untersuchst:

- Math.random()

- Array.splice(...)

- Element.className