

Übungsblatt 1 – Puzzle Game

Abgabe: 01.05.2020, 23:59 Uhr

Laden Sie eine ZIP-Datei mit dem Namen webscripting-ue01-ifxxbxxx.zip (zB webscripting-ue01-if20b001.zip) in Moodle (im Kurs „Web Scripting – Typescript UE01“) hoch, welche die zu erstellenden Dateien (und weitere benötigte Dateien – zB. Bilder) enthält. Die einfachste Variante ist den Visual Studio Code Projekt-Ordner in ein .zip Archiv zu packen und anschließend hochzuladen.

Ziel dieser Übung ist es, ein Bilder-Puzzle-Spiel mithilfe von TypeScript zu realisieren.

Erreichbare Punkte gesamt: 100

Grobe Spielbeschreibung:

- Es wird ein ausgewähltes (3 x 3) - Bilder-Puzzle angezeigt.
- Die einzelnen Bildausschnitte werden „durcheinander“ angezeigt.
- Durch Klicken auf Bildausschnitte werden diese vertauscht.
- Wird das Puzzle erfolgreich gelöst, wird das dem Spieler angezeigt.

1. JavaScript ist nicht verfügbar / benötigte Funktionalität nicht verfügbar

In modernen Browsern kann JavaScript explizit deaktiviert werden. Ist JavaScript nicht „verfügbar“, soll ein entsprechender Hinweis ausgegeben werden. Ist JavaScript „verfügbar“ soll das Spiel gestartet werden. Die Warnung, falls kein JavaScript verfügbar ist, soll zentriert am Bildschirm angezeigt werden. - 10 Punkte

~~2. Browser Detection~~

~~Ermitteln Sie den aktuell genutzten Browser via TypeScript. Geben Sie die Info oberhalb des Puzzles am Bildschirm aus (siehe Abbildungen). Ermitteln Sie den Typ dynamisch, d.h. wird die Seite in Chrome geöffnet, soll „Your browser is: Chrome“ ausgegeben werden. In Firefox soll „Your browser is: Firefox“ ausgegeben werden. Installieren Sie zumindest 2 unterschiedliche Browser (Chrome, Edge, Safari, Firefox, Opera, ...) und testen Sie die Funktion. - 10 Punkte~~

3. Spielfeld

Der Hintergrund der Seite soll einfarbig (nicht weiß!) sein. Das Spielfeld soll zentriert im Browser-Fenster positioniert sein. Wird die Seite geladen, soll der verwendete Browser als Text oberhalb des Spielfeldes angezeigt werden. Die einzelnen Bildausschnitte innerhalb des Spielfeldes sollen zufällig angeordnet sein (und als solche erkennbar sein -> siehe Abbildungen). Sie können als Basis zum „Mischen“ der Bildausschnitte folgenden Implementierung verwenden (gern aber auch eine andere): - 10 Punkte

```
function shufflePuzzleParts() {  
  const puzzleParts = [1, 2, 3, 4, 5, 6, 7, 8, 9];  
  let counter = puzzleParts.length;  
  while (counter > 0) {  
    const index = Math.floor(Math.random() * counter);  
    counter--;  
    const temp = puzzleParts[counter];  
    puzzleParts[counter] = puzzleParts[index];  
    puzzleParts[index] = temp;  
  }  
  return puzzleParts;  
}
```

4. Lösen eines Puzzles

Der Spieler kann durch Klicken einen Bildteil selektieren (siehe Abbildungen im Anhang), welcher dann hervorgehoben wird (zB. durch einen farbigen Rahmen, oder durch Verwendung von opacity). Die Selektion soll nicht zu einer Veränderung des Layouts führen („Verschiebung von Elementen“). Erst wenn der Benutzer einen zweiten Bildausschnitt anklickt, werden die beiden Bildteile vertauscht (und die Selektion aufgehoben). Eine Animation oder dgl. ist nicht notwendig. Klickt der Spieler auf das selektierte Bildteil erneut, soll die Selektion aufgehoben werden (= Rückgängig machen einer Selektion). Ist das Puzzle gelöst, soll kein weiterer Zug und keine Selektion mehr möglich sein und ein Text eine Erfolgsmeldung unter dem Puzzle angezeigt werden (siehe Abbildungen). - 20 Punkte

5. Allgemeines

Realisieren Sie Ihre Seite in HTML5. Schreiben Sie alle TypeScript-Funktionen und -Aufrufe in eine externe TypeScript-Datei. Die daraus erzeugte JavaScript Datei soll in index.html verlinkt werden.

- Verwenden Sie keine HTML-Tabellen als Layoutelemente für die Bildausschnitte.
- Die HTML-Seite soll beim Aufruf keine Elemente, welche für das Spiel notwendig sind, enthalten (Spielfeld, Bilder, ...). Nur der Hinweistext (siehe Punkt 1) und die „Browser Detection“ (siehe Punkt 2) dürfen in der HTML-Seite „statisch“ vorhanden sein (abgesehen vom HTML-Seiten-Grundgerüst).
- Verwenden Sie keine JavaScript-Bibliotheken.
- Verwenden Sie nicht innerHTML – verwenden Sie die DOM-Methoden zum Erzeugen/Entfernen von benötigten Elementen.

Ihre Lösung muss nicht pixelgenau den Abbildungen entsprechen. Die Abbildungen sollen dazu dienen die Aufgabenstellung zu verdeutlichen. Die grundsätzliche Funktionalität / das grundsätzliche Aussehen, wie in den Abbildungen, soll nicht verändert werden (Farben, Grafiken, Schriftarten und dgl. dürfen angepasst werden).



Abbildung 1 - Nach Start der Applikation

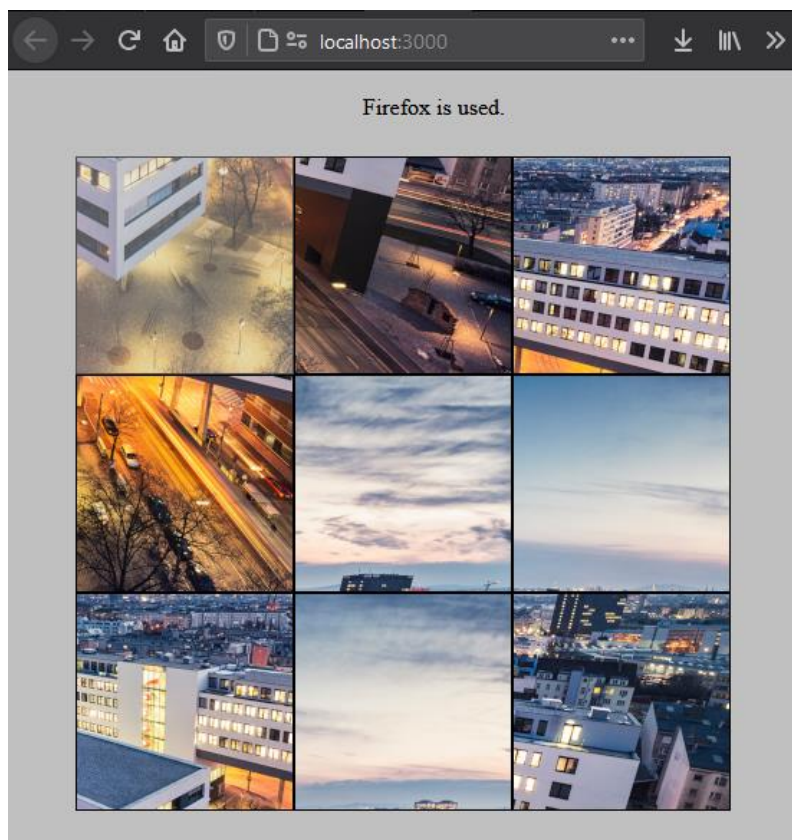


Abbildung 2 - Ein Element markiert



Abbildung 3 - Nachdem die ersten zwei Bildausschnitte getauscht wurden

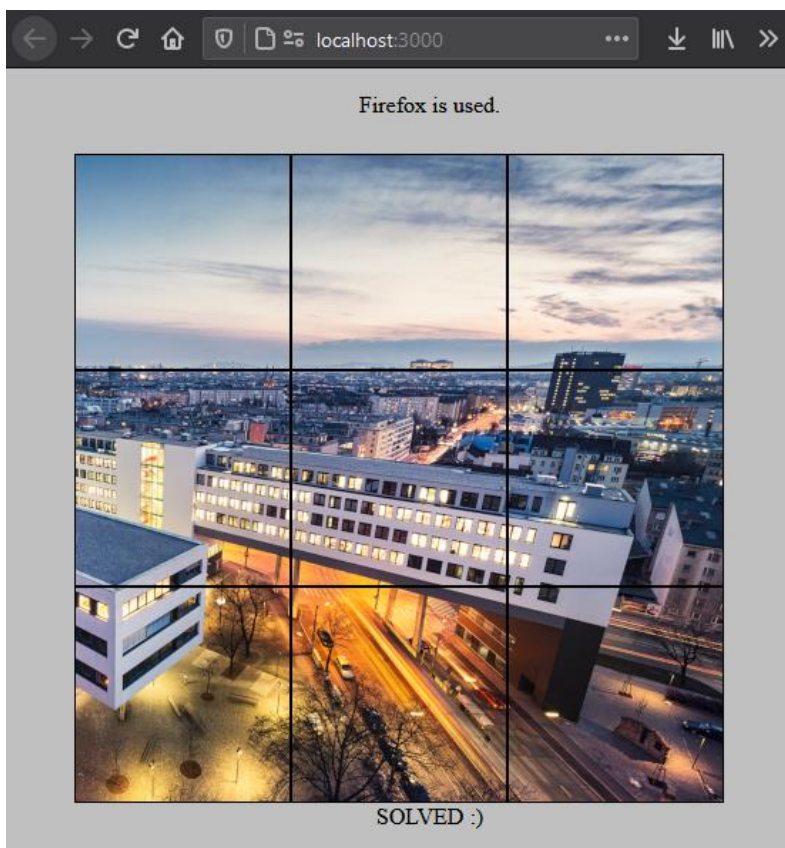


Abbildung 4 - Nachdem das Puzzle gelöst wurde