Inhaltsverzeichnis

Beschreibung HTML Aufbau

Die Seite enthält nicht viel, da das komplette Spiel mit allen Komponenten über Javascript erstellt wird. Somit wird nur ein als Container zum Einbinden des Spiels benötigt.

Stylesheets werden im Head eingebunden und die JS-Dateien am Ende des Bodys, damit diese erst nach laden der HTML Seite ausgeführt werden.

Beschreibung CSS-Dokumente

index.css

Diese Datei enthält das Styling für die Hauptseite. Da diese nicht viel Inhalt besitzt wird hier nur der Spielcontainer zentriert.

memory.css

Diese Datei enthält alle für das Spiel notwendigen Styles. Diese sind aufgeteilt in einen Abschnitt für die Spielkarten und einen für das Spiel.

Die Selektoren sind bewusst tief gewählt, um die Abhängigkeiten zu zeigen und Übersichtlichkeit zu wahren.

Beschreibung JS-Dokumente

index.js

Hier wird nur ein neues Memoryspiel erstellt und diesem der Container zugewiesen.

memory.js

Class Card

constructor(value, parent)

Der Konstruktor erwartet einen Wert, der der Karte zugeteilt wird. Dieser Wert wird als Unicodezeichen interpretiert und auf die Rückseite der Karte geschrieben.

Zudem muss die Klasse, welches das Objekt erstellt sich selbst mit übergeben damit die Karte, wenn sie geklickt wird, den Klickhandler der Elternklasse aufrufen kann.

Im Konstruktor wird des Weiteren der HTML Körper der Karte erstellt und gespeichert. Um das Anhängen muss sich jedoch das Elternobjekt kümmern.

Andere Funktionen

„flip()“

> Fügt der Karte die Klasse „turn“ hinzu oder entfernt diese wieder.

„hide()“

> Fügt der Karte die Klasse „hidden“ hinzu oder entfernt diese wieder.

„handleClick()“

> Ist der Clickhandler der Karte. Dieser ruft den Clickhandler der Elternklasse sollte dieser nicht blockiert sein.

Class Memory

Variablen

Die Spieler werden als Objekte innerhalb des Arrays *„playerList“* abgespeichert. Sie enthalten Informationen zum Punktestand aber auch den Listeneintrag, welcher Spielerinformationen während dem Spiel anzeigt. Welcher Spieler gerade aktiv ist wird als Index für die Spielerliste in der Variablen *„activePlayer“* abgespeichert.

Die Spielkarten Objekte werden im Array *„cards“* abgespeichert. Um die ausgewählten Karten miteinander vergleichen zu können wird die erste Karte, die der Spieler anklickt in „firstCard“ abgespeichert und die Flagvariable *„firstTry“* gibt an, ob die abgespeicherte Karte die ist die zuerst angeklickt wurde.

Die vergangene Zeit zum Spielbeginn wird in *„passedTime“* abgespeichert, das zum Zeit hochzählen benötigte Intervall in *„timerInterval“*.

Erstellung des HTML Konstruktes

Um die Erstellung des Spiels übersichtlich zu halten, ist die Erstellung auf Hilfsfunktionen verteilt. Diese sind mit dem Präfix *„createHTML“* gekennzeichnet. Deren Aufruf findet lediglich einmal im Konstruktor statt, da das Grundgerüst ja nicht öfter erstellt werden muss. Einige HTML Objekte in Variablen abgespeichert, um spätere Zugriffe zu beschleunigen. Sämtliche HTML Objekte werden als solche mit dem Präfix *„html_“* gekennzeichnet.

Spielerinteraktion

Folgende Funktionen können direkt durch den Spieler über Buttons ausgelöst werden:

„addPlayer()“

> Wird ausgelöst durch den Button zum Hinzufügen eines neuen Spielers. Die Funktion erstellt einen neues Spieler Objekt und dazugehörende Listeneinträge, um diesen Spieler für den Benutzer sichtbar zu machen. Es können maximal 4 Spieler existieren, sollte die Funktion danach aufgerufen werden macht sie nichts.

„remPlayer()“

> Wird ausgelöst durch den Button zum Entfernen eines Spielers. Löscht den letzten Spieler aus der Liste und sämtliche Verbindungen. Sollte lediglich ein Spieler existieren wird dieser nicht gelöscht, jedoch sein Name zurückgesetzt.

„startGame()“

> Wird ausgelöst durch Klicken auf den Startknopf. Sämtliche vom Spieler vorgenommen Einstellungen werden übernommen und das Interface verändert. Der Status des Spiels ist nun aktiv. Die Karten werden erstellt und dem Spielfeld angehängt.

„resetGame()“

> Wird aufgerufen durch den Resetknopf oder am Ende vom Spiel über *„New Game“*. Die Funktion sorgt dafür, dass obgleich das Spiel ordnungsgemäß beendet wurde, das Spiel in den Startzustand zurückversetzt wird. Vergebene Spielernamen und Spieleinstellungen werden jedoch beibehalten.

„handleClick()“

Wird durch den Klick auf eine Karte aufgerufen und enthält den Hauptbestandteil der Spiellogik. Hier wird die erst angeklickte Karte abgespeichert, und die Zweite mit der Ersten verglichen. Sollten sie nicht übereinstimmen werden beide Karten wieder verdeckt und der nächste Spieler ist an der Reihe. Falls sie jedoch übereinstimmen, werden dem Spieler die Punkte gutgeschrieben und er darf es noch einmal versuchen. Sollten alle Karten gefunden worden sein wird das Spiel durch den Funktionsaufruf von *„endGame()“* beendet.