UE2 – Client-seitiges JavaScript, jQuery, SVG (25 Punkte)

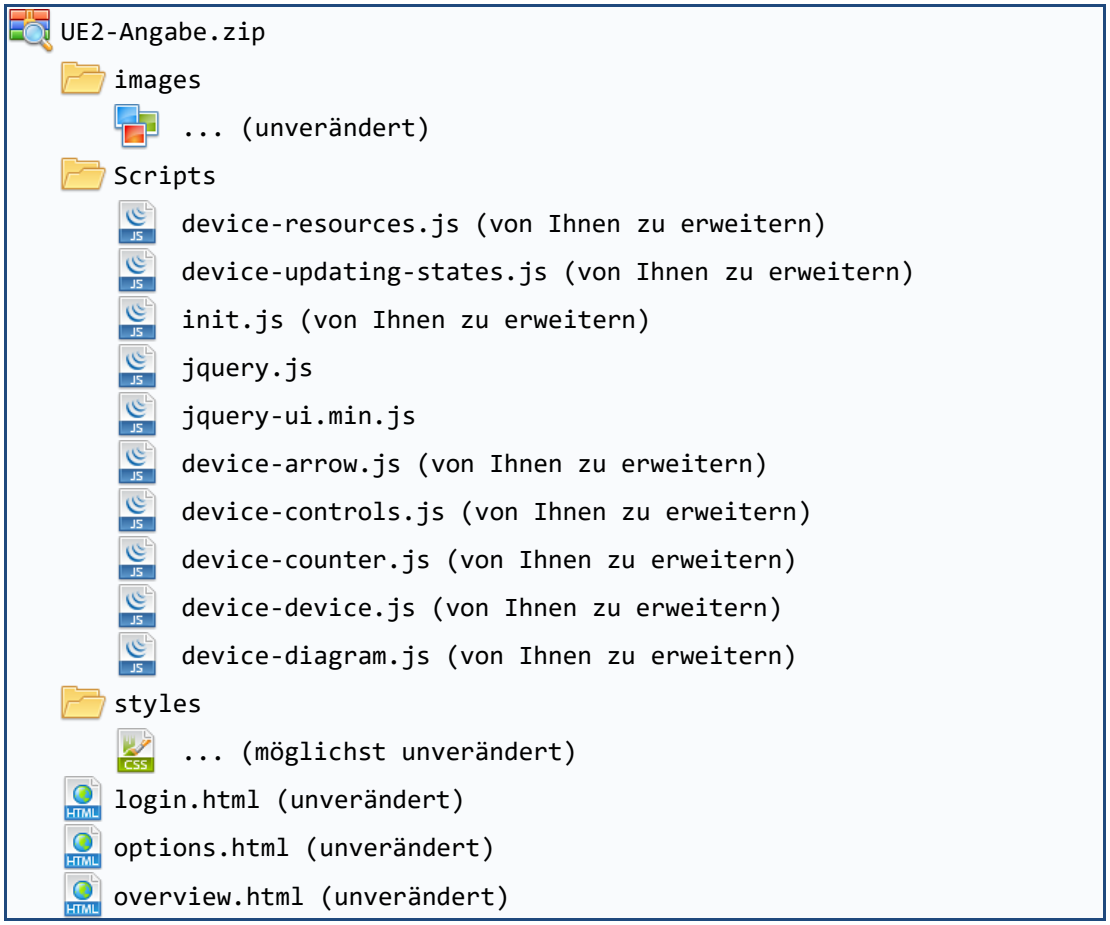
Ziel dieses Übungsbeispiels ist es, clientseitige Technologien kennenzulernen und einzusetzen. Dazu soll das Beispiel aus Übung 1 als Grundlage dienen. Die dort manuell platzierten Geräte im Produktionssystem sollen nun dynamisch über JavaScript positioniert werden können. Zusätzlich zu der Drag & Drop Funktionalität sollen mit Hilfe von JavaScript und in weiterer Folge mit jQuery einfache Animationen auf Scalable Vector Grafiken (SVG) durchgeführt werden und Verbindungen durch Pfeile zwischen den Geräten innerhalb des Produktionsketten-Diagramms erstellt werden können.

Deadline der Abgabe via TUWEL: **Sonntag, 22.04.2018 23:55 Uhr**

Nur ein Gruppenmitglied muss die Lösung abgeben!

Angabe

Diese Angabe umfasst folgende Dateien:



Bitte beachten Sie, dass nicht alle der bereitgestellten Dateien bearbeitet werden sollen (siehe *„Abgabemodalität“*). Nehmen Sie jedoch keinesfalls Änderungen an den verwendeten Bibliotheken vor. Alle für diese Aufgabe benötigten Bibliotheken sind bereits enthalten und entsprechend eingebunden und bedürfen daher kein Einschreiten Ihrerseits.

Anmerkungen zum Datenmodell & Ablaufmodell

Für diese Übung gibt es 5 Klassen (model-*.js), eine Datei für die Initialisierung aller Komponenten (init.js), eine Datei zur Speicherung der SVGs (device-resources.js) und eine Datei zum Zeichnen und Updaten der Geräte innerhalb des Diagramms (device-updating-states.js).

Die 5 Klassen repräsentieren jeweils eine Komponente des Ablaufmodells der Overview Seite.

- Mithilfe der Klasse `Arrow` werden gezeichnete Pfeile innerhalb des Diagramms gespeichert und verwaltet.
- Mithilfe der Klasse `Controls` werden die Kontrollelemente zum Verändern der Gerätezustände verwaltet.
- Mithilfe der Klasse `Counter` werden die Zähler für Anzahl der Geräte und Pfeile innerhalb des Diagramms gespeichert sowie verändert.
- Mithilfe der Klasse `Device` werden die Geräte innerhalb des Diagramms gespeichert und verwaltet.
- Mithilfe der Klasse `Diagram` wird das Diagramm, auf dem die Produktionskette gezeichnet wird, verwaltet. Weiters werden hier alle Geräte und Pfeile gespeichert und verwaltet.

Zu beachten sind die Eingabeparameter der jeweiligen Klassen.

Klassen, Methoden und Variablen sind mit JSDoc[1] dokumentiert.

In den Angabe-Dateien befinden sich einige ***TODOs***, an die Sie sich bei der Ausarbeitung halten können aber nicht müssen.

Hinweis:

Sie haben grundsätzlich die Möglichkeit, alle vorgegebenen Klassen, Methoden und Variablen so abzuändern, wie es für Ihre Implementierung geeignet ist. Die generelle Struktur sollte jedoch, wie in der Abgabemodalität zu sehen ist, eingehalten werden.

Hauptanforderungen an Ihre Implementierung

- **Initialisierung:**
  Relevante Dateien: init.js
  a) Zu erstellen sind Objekte der Klasse Counter für die Zähler von Geräten und Pfeilen oberhalb des Produktionsketten-Diagramms (siehe *„Zähler für Elemente (Geräte & Pfeile) im Diagramm“* weiter unten).
  b) Zu erstellen ist ein Objekt der Klasse Controls. Übergeben Sie mithilfe von jQuery das Form der Kontrollelemente (ansprechbar über die ID `controls`) an diese Klasse.
  c) Zu initialisieren ist die Klasse Diagramm. Übergeben Sie dieser Klasse die erstellen Counter-Objekte und das Objekt der Kontrollelemente aus a) bzw. b).
  d) Zu initialisieren sind alle Geräte der Sidebar. Siehe nächsten Punkt zu Drag & Drop.

1 http://usejsdoc.org

- ***Drag & Drop:***
  Relevante Dateien: init.js, model-diagram.js, …
  a. Implementieren Sie mithilfe des jQuery UI Draggable und Droppable Plugins (jquery-ui.js ist bereits in overview.html eingebunden) eine Drag & Drop Funktionalität für die verfügbaren Geräte der Sidebar, um diese in das Diagramm ziehen zu können. Setzen Sie auf die Geräte der Sidebar Draggable und auf das Diagramm Droppable (in init.js oder model-diagram.js).

- ***Hinzufügen der Geräte zum Diagramm:***
  Files: model-diagram.js, model-device.js, device-resources.js, …
  a. Nach dem „Droppen" von Geräten auf das Diagramm sollten neue HTML Container für diese Instanzen erstellt werden. Dieser Container enthält wie bei Übung 1 „Vorgänger", „Nachfolger", den Titel des Gerätes (Gerätetyp + fortlaufende Nummer) und das SVG des Gerätes. Fügen Sie den Container im HTML DOM in die Liste `<ul class="devices device-list"></ul>` hinzu.

- ***Eigenschaften der Geräte innerhalb des Diagramms:***
  Relevante Dateien: model-diagram.js, model-device.js, …
  a. Auf den hinzugefügten/gezeichneten Geräten im Diagramm sollte wiederum die Drag&Drop Funktionalität hinzugefügt werden. Es ist zu beachten, dass die Geräte nicht außerhalb des Diagramms verschoben werden können. Das Überlappen von Geräten ist nicht relevant.
  b. Weiters sollten die Geräte über Events anklickbar gemacht werden. Wurde ein Geräte ausgewählt, sollte auf das Gerät die CSS Klasse `active` in style.css gesetzt werden.
  c. Wenn die Maus über ein Gerät bewegt wird, sollte der Steuerungspfeil zum Zeichnen eines Pfeils (Verbindung) innerhalb an der rechten unteren Ecke des Gerätes sichtbar werden. Hierfür ist das HTML Element mit der ID `arrow-device-add-reference` (siehe overview.html) zu verwenden und innerhalb des hinzugefügten Gerätes zu platzieren (entweder als Kopie oder verschoben).
  d. Weiters sollte das Kontextmenü mit der Klasse `contextMenu` (siehe overview.html) bei einem Rechtsklick auf ein Gerät an der Position des Mausklicks angezeigt werden.
  e. Detailseite: Über einen Doppelklick auf ein Gerät bzw. über das Klicken des Buttons „Detailseite" des Kontextmenüs sollte über ein JavaScript `alert()` der Titel des Gerätes (samt fortlaufender Nummer) ausgegeben werden. Dies dient als Platzhalter für die in dieser Übung noch nicht zu implementierende Detailseite.
  f. Löschen: Hinzugefügte Geräte sollten wieder aus dem Diagramm gelöscht werden können. Die fortlaufende Nummer der Geräte wird dadurch nicht verändert – es entstehen lediglich Lücken. Das Löschen sollte durch Anklicken des jeweiligen Gerätes und anschließendem Drücken der Taste „ENTF" oder über den „Löschen"-Button des Kontextmenüs möglich gemacht werden. Zusätzlich sollten alle mit dem zu löschendem Gerät verbundenen Pfeile mitgelöscht werden.

- ***Hinzufügen von Pfeilen (Verbindungen):***
  Relevante Dateien: model-diagram.js, model-device.js, model-arrow.js, …
  a. Die Geräte sollten über Pfeile miteinander verbunden werden können. Um einen Pfeil von einem Gerät zum anderen Gerät ziehen zu können, muss das Zeichnen eines Pfeiles aktiviert werden. Hierfür gibt es mehrere Möglichkeiten, die zu implementieren sind: durch Klicken und Aktiv-Schalten des Pfeil-Buttons der Sidebar (siehe ID `arrow-sidebar-add` in overview.html); durch Klicken des Pfeil-Buttons innerhalb des Geräte (siehe vorheriger Punkt); durch Drücken der Taste „a“/“A“. Bei allen drei Möglichkeiten sollte der Pfeil-Button in der Sidebar während dem Zeichnen des Pfeils auf „Aktiv“ geschaltet werden. Verwenden Sie hierfür die Klasse `active` in style.css.
  b. Das Zeichnen der Pfeile sollte nur innerhalb des Diagramms möglich sein.
  c. Die Pfeile sollten zwischen den Geräten so gezeichnet werden, dass der Startpunkt des Pfeils sowie der Endpunkt des Pfeils jeweils vom Mittelpunkt ausgeht, jedoch im Diagramm erst ab den Schnittpunkten des Startgeräts bzw. des Endgeräts sichtbar ist. Hierfür sollte die Methode `getIntersectionCoordinates()` in model-device.js verwendet werden. Diese berechnet den Schnittpunkt des Gerätes unter Berücksichtigung des Mittelpunkts des Endgerätes (`targetPosition`) bzw. während dem Zeichnen der Mausposition.
  d. Bereits gezeichnete Pfeile (Verbindungen) sollten ausgewählt und gelöscht werden können. Zum Auswählen soll die Klasse `active` auf den Path des Pfeils gesetzt werden. Das Löschen sollte über durch das Drücken der Taste „ENTF“ möglich sein. Es ist dabei zu beachten, dass die eingetragenen Verbindungen an den Geräten (sprich den Endpunkten des Pfeils) ebenfalls gelöscht werden.

- ***Zähler für Elemente (Geräte & Pfeile) im Diagramm:***
  Relevante Dateien: model-counter.js, …
  a. Oberhalb des Diagramms befinden sich 3 Zähler (Counter), die die Anzahl der im Diagramm befindlichen Geräte, Pfeile und im Prozess erstellen Produkte angeben. Für jeden Zähler sollte ein Counter (siehe model-counter.js) erstellt werden, welcher durch Hinzufügen von neuen Geräten oder Pfeilen erhöht bzw. durch Löschen von Geräten oder Pfeilen vermindert wird. Der Counter für die Anzahl der Produkte ist in dieser Übung noch nicht zu beachten.

- ***Kontrollelemente:***
  Files: model-controls.js, device-updating-states.js, …
  a. Verwalten Sie die gezeichneten Geräte innerhalb des Diagramms in der Klasse Controls. Fügen Sie beim „Droppen“ eines Gerätes der Sidebar innerhalb des Diagramms das neu zu zeichnende Gerät einer Liste in Controls hinzu. Entfernen Sie das Gerät von der Liste, wenn das Device gelöscht wird.
  b. Die Geräte innerhalb des Diagramms verfügen über mehrere unterschiedliche Zustände, welche über die Kontrollelemente unterhalb des Diagramms in dieser Übung gesteuert werden können. Zum Ändern der Zustände sollten die Methoden aus device-updating-states.js verwendet werden. In diesem File befinden sich leere Methoden, die von Ihnen ausgefüllt und zum Ändern der Zustände verwendet werden sollen. Mittels jQuery oder

Standard-JavaScript können Sie auf IDs bzw. Klassen innerhalb der SVGs der Geräte zugreifen und diese verändern.

c. Die entsprechende Update-Methode (je nach Gerätetyp) wird dem Konstruktor von `Device` übergeben und soll in `Device.updateDevice()` aufgerufen werden. Aus der Klasse `Controls` sollen alle bereits existierenden Geräte aktualisiert werden, wenn das Formular abgesendet wird.

d. Wird ein neues Gerät zum Diagramm hinzugefügt, sollte dies bereits den eingestellten Zustand des Kontrollelements übernehmen und dementsprechend gezeichnet werden.

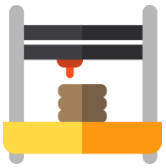
Die Zustände sollen wie folgt aussehen:

3D-Drucker

Unfertig

Teilweise fertig

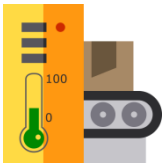
Fertig



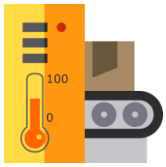
Maschine

Das Thermometer soll sich für jeden Wert ändern.
Die Farben sind an die Intervalle anzupassen.

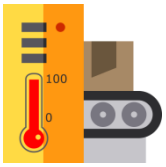
Untergrenze - 25° - 33°



34° - 50° - 66°

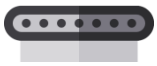


67° - 100° - Obergrenze



Förderband

Aus

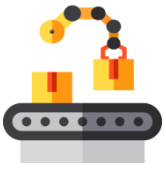


Ein

Intelligentes Förderband

Aus

Ein

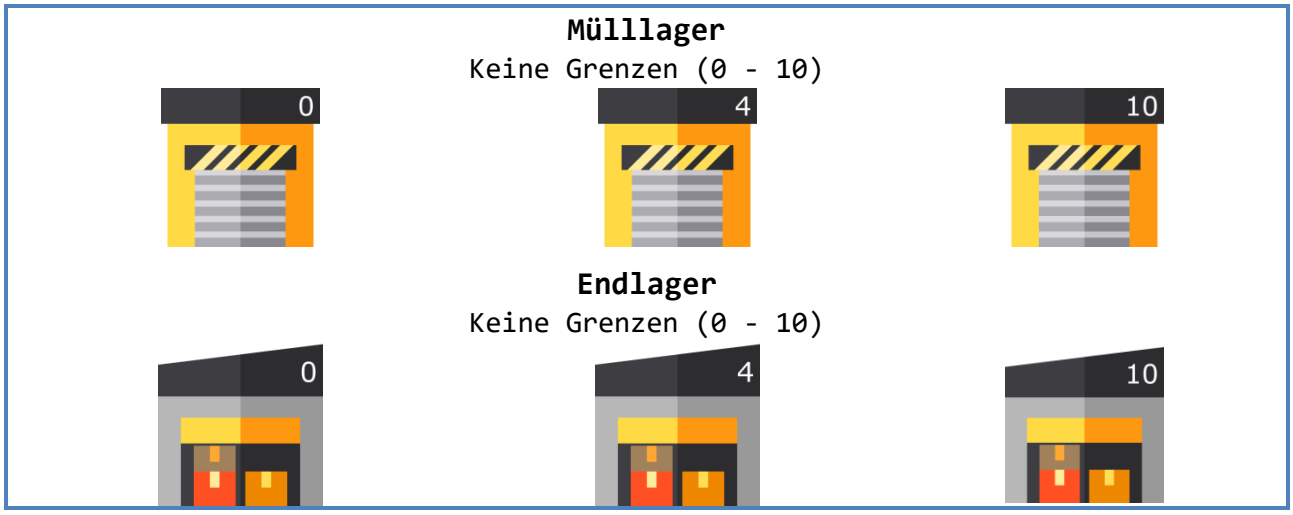


Temporäres Lager

0

1-4

5 - 10



- ***Bonuspunkte (3 Punkte):***
  Relevante Dateien: model-diagram.js, model-device.js, …
  Sie haben die Möglichkeit, bis zu 3 Bonuspunkte zu erhalten, wenn Sie folgende Funktionalitäten hinzufügen => Zeichnen von Verbindungen zwischen Geräten durch Tastatursteuerung:
  a) Über die Taste „Tab“ kann zwischen den einzelnen Geräten und Pfeilen innerhalb des Diagramms sichtbar gewechselt werden können.
  b) Durch Drücken der Taste „Enter“ wird ein Gerät (das über „Tab“ gerade ausgewählt wurde) ausgewählt, von welchem aus ein Pfeil zu einem anderen Gerät gezeichnet werden können soll. Drückt man anschließend auf die Taste „a“/“A“, so wird der Zeichenmodus für Pfeile auf „Aktiv“ geschaltet und ein Pfeil ausgehend von dem ausgewählten Gerät gezeichnet. Wechselt man nun mit „Tab“ das Gerät und drückt anschließend wieder „Enter“, soll eine Verbindung (Pfeil) zwischen den beiden Geräten erzeugt werden.

  Achtung: Um die Bonuspunkte erhalten zu können, erstellen Sie bitte ein ***README.txt*** im Abgabe-Ordner und beschreiben Sie, ob Sie a) oder b) bzw. beide implementiert haben.

Hinweise

Ausführen der Applikation

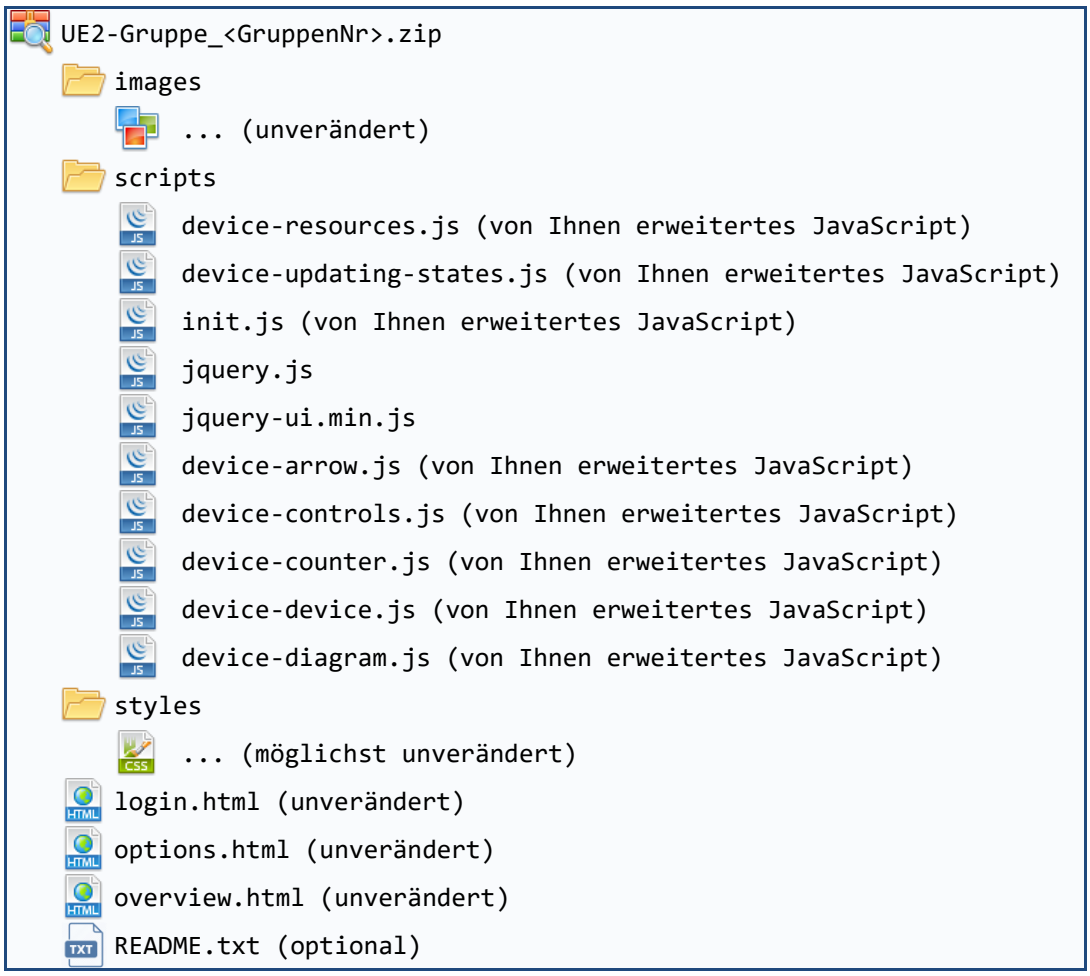
Die Applikation sollte OHNE Webserverumgebung über das File-Protokoll[2] ausgeführt und angezeigt werden. Daher ist es nicht möglich, Bilder, SVGs oder andere Elemente über AJAX oder ähnliche Technologien zu laden. Aus diesem Grund sind die zu verwendenden SVGs für die Geräte innerhalb des Produktionsketten-Diagramms im File `device-resources.js` hard-coded.
Für diese Übung darf jede JavaScript Version (ES5, ES6, ES7) verwendet werden, sofern die Browser-Kompatibilität gegeben ist.

Validierung

Das W3C stellt online sowohl HTML-Validatoren https://validator.w3.org als auch CSS-Validatoren http://jigsaw.w3.org/css-validator zur Verfügung. Weitere Validierungsoptionen entnehmen Sie der Angabe von Übung 1.

Abgabemodalität

Beachten Sie die allgemeinen Abgabemodalitäten des TUWEL-Kurses. Zippen Sie Ihre Abgabe, sodass sie die folgende Struktur aufweist:



2 https://en.wikipedia.org/wiki/File_URI_scheme

- Alle Dateien müssen UTF-8 codiert sein!
- Nur ein Gruppenmitglied muss die Lösung abgeben!
- Wird das Abgabeschema bzw. der Name der Zip-Datei (UE2-Gruppe_<GruppenNr>.zip) nicht eingehalten, kommt es zu Punkteabzügen!