

Aufgabenblatt 6

Webtechnologien
Wirtschaftsinformatik
WS 2020/21

Lernziele

Es wird folgendes gelernt:

- Funktionsweise von Ajax
- Kommunikation zwischen Browser und Server
- Realisierung einer Single-Page-Anwendung

JavaScript, Ajax und <div> - 3 Punkte

Nun wird das vielleicht bekannte Ziegenproblem simuliert (siehe Liste mit Links unten): Dabei steht der Kandidat vor drei geschlossenen Türen. Hinter zwei dieser Türen befindet sich eine Ziege als Symbol für eine Niete, hinter der dritten Tür ein Auto als Symbol für den Gewinn. Nun wählt der Kandidat eine Tür, die sich nicht öffnet. Der Spielleiter öffnet dann eine andere Tür mit einer der beiden Ziegen. Nun kann der Kandidat seine Entscheidung ändern oder bei der früher gewählten Tür bleiben. Anschließend wird die endgültig gewählte Tür geöffnet und der Kandidat hat gewonnen oder nicht.

Das Problem hierbei liegt darin: Sollte der Kandidat nach dem Öffnen der Tür mit der Ziege durch den Spielleiter seine Entscheidung ändern oder dabei bleiben? In den Links unten stehen einige Analysen zur Auswahl.

Sie sollen dieses Quiz mit Ajax realisieren. Dazu wird die Seite einmalig geladen und bleibt während eines Spiels im Browser bestehen (Schritte 2 und 3 unten). Der Zug des Spielers wird per Ajax dem Server mitgeteilt. Alle Ausgaben werden mit Modifikation des DOMs realisiert. Machen Sie dazu am Anfang einen Layout-Entwurf mit sämtlichen Elementen aus allen Phasen des Ziegenspiels mittels Pencil oder einem anderen Werkzeug. Dieser Entwurf gehört zur Abgabe.

Als nächstes entwerfen Sie die Kommunikationsnachrichten, also das Protokoll. Der Server übernimmt die Rolle des Spielleiters, der Surfer ist der Kandidat. Der Server fertigt eine Statistik an, um festzustellen, welche der beiden Strategien für den Kandidaten die bessere ist. Damit dies über mehrere Spiele geht, muss mit Sessions gearbeitet werden.

Der folgende Ablauf definiert die Anfragen seitens des Browsers am Server,

wobei alle Parameter mit einem POST-Request in der URL-Form (und Kodierung) mitgeteilt werden (es werden also keine JSON-Strings etc. benötigt):

1. Zum Starten wird die URL des Quiz in die Adresszeile des Browsers eingegeben.

Hinweg: kein Parameter

Rückweg: HTML/CSS/JS für ein neues Spiel

2. Senden der Wahl der Tür auf dem Hinweg und Empfangen der Wahl des Servers (Spielleiters) bzgl. der Ziege auf dem Rückweg. Im Browser wird die entsprechende Tür mit einer Ziege geöffnet.

Hinweg: choice=<ID> mit Ajax

Rückweg: open=<ID> mit Ajax

<ID> ist eine Nummer zur Identifizierung einer Tür, z.B. 0 bis 2.

3. Endgültige Entscheidung des Kandidaten. Auf dem Rückweg wird dem Spieler mitgeteilt, ob seine Entscheidung richtig war. Im Browser werden nun alle Türen geöffnet und die Ziegen bzw. das Auto gezeigt.

Hinweg: open=<ID> mit Ajax

Rückweg: success=<Bool> mit Ajax

<Bool> bedeutet True (1) oder False (0).

4. Der Spieler drückt den Knopf "Neues Spiel". Im Server wird dann die Statistik fortgeschrieben und ein neues Spiel vorbereitet.

Hinweg: kein Parameter

Rückweg: HTML/CSS/JS für ein neues Spiel (es wieder von vorn los)

5. Der Spieler drückt den Knopf "Ende".

Hinweg: kein Parameter

Rückweg: Ausgabe der Statistik, wie viele Male die Ziege bzw. das Auto gewählt wurde

<ID> ist die Nummer der Tür (0, 1, 2), <Bool> ist entweder 1 oder 0, wobei 1 Gewinn und 0 Niete bedeutet. success=1 bedeutet, dass der Kandidat richtig bzw. bei 0 falsch gewählt hat.

In den Schritten 1, 4 und 5 werden einfach neue Seiten geladen, während bei 2 und 3 mit Ajax Daten von Server geholt werden, die im Browser verarbeitet werden. Diese beiden Schritte bilden das eigentliche Spiel.

Bitte beachten Sie, dass im dritten Schritt der Fehler auftreten kann, dass der Spieler eine schon geöffnete Tür wählt. Hier müssen Sie einen Fehlertext ausgeben. Dies erfolgt allein im Browser; dazu benutzen Sie am besten den Mechanismus der Fehlerausgabe aus dem Arbeitsblatt über die Formularprüfung.

Da mehrere Klicks bzw. Requests zu einer Session gehören, müssen Sie auf der Serverseite mit Sessions arbeiten.

Und nun die einzelnen Schritte:

1. [Abgabe] Sie entwerfen die Seite mit den Layout-Angaben der Layer und eine CSS-Datei – also wie in den vorherigen Aufgabenblättern. Vergessen Sie nicht den Knopf für ein neues Spiel und für die Beendigung.
2. Zur Seite gehören auch Graphiken: Erstellen Sie eine Graphik für (a) geschlossene Tür, (b) Ziege und (c) Auto. Alle drei Graphiken haben exakt dieselbe Größe. Initial wird (a) angezeigt. Das Öffnen der Tür realisieren Sie durch Austausch der Graphik. Dazu schreiben Sie sich eine kleine JS-Routine, siehe dazu den Foliensatz der Veranstaltung.

Wenn Sie an die Graphik ein onClick-Ereignis mit dem Aufruf des JS-Routine binden, können Sie das Austauschen wie folgt testen: Um die Türen zu unterscheiden, rufen Sie eine JS-Routine mit der Nummer der Tür auf, die dann die Routinen zum Tauschen der Graphiken aufruft. Das Austauschen von Bildern ist im JavaScript-Foliensatz als Beispiel beschrieben.

3. Jetzt konzipieren Sie die Ajax-Kommunikation: konkret wer mit welcher ID was zum Server schickt und was als Antwort erhält.
4. Nun implementieren Sie eine allgemeine Ajax-Routine – als Objekt oder als Sammlung von Routinen, die allgemein die Kommunikation realisieren. Hierbei können Sie sich auch an den Foliensätzen über JavaScript orientieren.
5. Nun kommt die Logik des Ablaufs im Browser in den Schritten 2 und 3 dran; hierzu benötigen Sie globale Variablen für den Zustand des Dialogs und eine größere Routine, die jeweils an die Graphiken per Ereignis gebunden wird und die die Kommunikation mit dem Server steuert.

Das gesamte JavaScript-Teil wird in eine Datei geschrieben, die in die HTML-Seite am Ende eingebunden wird.

6. [Abgabe] Jetzt müsste das Quiz fertig sein; es hat nur eine karge Oberfläche und das wird nun geändert. Vor jedem Spielschritt wird in einem reservierten Bereich (<div>) ein auffordernder Text, z.B. "Bitte wählen Sie eine Tür" bzw. die Fehlermeldung "Diese Tür kann nicht gewählt werden" sowie das Ergebnis: "Sie haben gewonnen" oder "Ooohh, Sie haben leider verloren". Wie das geht, haben Sie im Arbeitsblatt über die Formularprüfung schon kennen gelernt.

Das ist jetzt schon eine richtige Web-Anwendung.

Hinweis zu den Sessions: Wenn die PHP-Session nicht gehen sollten, dann müssen Sie den Session-Mechanismus von Hand durch Serialisierung/Deserialisierung wie in einer der vorherigen Aufgaben realisieren.

Hinweise und Links

- <https://de.wikipedia.org/wiki/Single-page-Webanwendung>
- Ajax und DOM - der Foliensatz über JavaScript
- <https://de.wikipedia.org/wiki/Ziegenproblem>
- <https://www.youtube.com/watch?v=F2a0LE8Ffj4>
- <http://www2.hs-fulda.de/~grams/hoppla/wordpress/?p=526>
- <https://www.zeit.de/2004/48/N-Ziegenproblem>
- http://www.sylviarichter.de/images/uni/ausarbeitung_ziegenproblem.pdf
- https://praxistipps.chip.de/ziegenproblem-die-loesung-einfach-erklaert_98889

Abnahme und Bewertung

Das Ergebnis des Schrittes 6 wird per Email abgegeben. Dabei muss die gesamte Site als zip-Datei so wie sie unter htdocs liegt bzw. das netbeans-Projekt abgegeben werden. Bei Abgaben per Email muss die Lösung auf den Rechnern der Dozenten im ersten Anlauf laufen.

Der Entwurf des Layouts gehört zur Abgabe.

Es sind max. 3 Punkte zu erreichen; fehlt etwas oder ist es nicht in Ordnung, werden Punkte abgezogen bzw. die Lösung nicht anerkannt. Auch ist es nicht erlaubt, Frameworks wie z.B. jQuery oder Bootstrap zu benutzen.

Es müssen die in den vorherigen Aufgabenblättern vorgestellten Regeln über CSS, Layout und HTML4/5 eingehalten werden.