

W3WI_110.2 - Web-Entwicklung

Dozent: Prof. Dr. Michael Eichberg
Kontakt: michael.eichberg@dhbw.de, Raum 149B
Version: 24EG/EH

Ausgewählte Inhalte gem. MHB

Kerninhalte

- HTML, CSS, JavaScript als clientseitige Web-Technologien und aktuelle APIs (z.B. HTML5 und verwandte Technologien)
- Übertragungsprotokolle und APIs zwischen Client und Server (z.B. HTTP, HTTPS, WebSockets, XMLHttpRequest, Fetch API)
- Kommunikation zwischen einzelnen Komponenten Web-basierter Anwendungen
- Optimierung von Webseiten für verschiedene Zielsysteme

Prüfungsleistung - Portfolio

Hintergrund

- das Modul hat 55 VL
- Web-Programmierung hat 33VL und geht mit **70** von 120 Modulpunkten ein

Portfolioaufgabe

Entwicklung eines kleinen *verteilten* zwei Personen Spiels mit HTML, CSS und JavaScript.

Der Einsatz von großen Frameworks ist nicht erlaubt. Der Einsatz von Socket.io und Express (auf Serverseite) ist erlaubt, um die Kommunikation zwischen den Clients zu vereinfachen. Der Einsatz weiterer Frameworks ist nur nach Absprache gestattet. Der Einsatz von Frameworks zum Testen (z. B., Jest) ist erlaubt.

Mögliche Spiele

Jeder muss ein anderes Spiel wählen!

- | | |
|--|---|
| <input type="checkbox"/> Kalaha | <input type="checkbox"/> Stadt-Land-Fluss |
| <input type="checkbox"/> 4 Gewinnt | <input type="checkbox"/> Hangman |
| <input type="checkbox"/> Halma | <input type="checkbox"/> Käsekästchen |
| <input type="checkbox"/> Dame | <input type="checkbox"/> Reversi |
| <input type="checkbox"/> Mühle | <input type="checkbox"/> Stein-Schere-Papier |
| <input type="checkbox"/> Schiffe versenken | <input type="checkbox"/> Sprouts (Punkte mit Linien verbinden, ohne Überkreuzungen) |
| <input type="checkbox"/> Memory | <input type="checkbox"/> Hex (Sechseckflächen verbinden) |

Spielablauf

1. Ein Spieler geht auf eine Login-Seite, um sich beim Server anzumelden
2. Sobald sich ein weiterer Spieler anmeldet, werden diese beiden Spieler automatisch verbunden und ein Spiel wird gestartet
3. Die Spieler machen abwechselnd Ihre Züge
D.h. der Client sendet eine Nachricht an den Server, um seinen Zug zu machen, daraufhin wird der Server die Nachricht an den anderen Spieler weiterleiten.
4. Sobald ein Spieler gewonnen hat, wird dies beiden Spielern angezeigt und das Spiel beendet
5. Die Spieler können sich erneut anmelden

Anforderungen und Rahmenbedingungen

- das Spiel muss auf Firefox, Safari und Chrome in der jeweils aktuellen Version lauffähig sein
- das Spiel muss ab einer Auflösung von 1024x768 Pixeln (Desktop) spielbar sein
- kommt es zu einem Fehler (z.B. Verbindungsabbruch), so wird das Spiel einfach terminiert; der Server darf nicht abstürzen
- die Serveranwendung läuft auf dem bereitgestellten Server

Bewertung (max. 70 Punkte)

Bestandteile

Codepräsentation - HTML + CSS: 20 Punkte (max.)

Codepräsentation - JavaScript: 20 Punkte (max.)

Qualität des finalen Codes: 20 Punkte (max.)

Läuft das Spiel: 10 Punkte (max.)

Kriterien für die Bewertung des Codes

Möglichkeiten für Bonuspunkte

Responsiveness: max 5 Punkte (d. h. die Anwendung passt sich an verschiedene Bildschirmgrößen an, z.B. Desktop, Tablet, Smartphone)

Internationalisierung:
max 5 Punkte (d. h. Unterstützung von mind. 2 Sprachen wird geboten)

Cheatingprävention:
max 5 Punkte (d. h. Maßnahmen zur Verhinderung von Betrug, z.B. durch Validierung von Eingaben, Überprüfung von Spielständen)
Bitte **dokumentieren Sie dies ggf. sehr deutlich** im Code!

Ablauf

27.8. - 1.10.:

Vorlesungen mit Übungen bzgl. HTML, CSS und JavaScript.

8.10: Individuelle Hilfe bei der Entwicklung

15.10.: Code Präsentationen/Diskussion - HTML und CSS

Pro Person 9-10 Minuten - Aufteilung ca. 50/50; max 40/60.

Abgabe ist bereits eingerichtet (Deadline siehe Moodle!). Es kann ein Zip-Archiv und ggf. ein PDF hochgeladen werden.

Bewertungskriterien bei der HTML/CSS Präsentation

Es ist insbesondere darzustellen wie das Layout auf HTML Elementebene erstellt wurde (mit Tabellen, Canvas, DIVs,...) und warum dies eine geeignete Lösung für das Spiel ist.

Bzgl. des CSS ist insbesondere die Strukturierung/Modularisierung darzustellen und wie ein konsistentes Layout (über Browserversionen hinweg) erreicht wurde. Darüber hinaus ist insbesondere der Code, der mit dem Layout in Verbindung steht (display, position, margin, padding etc. Eigenschaften) zu erläutern.

22.10.: Code Präsentationen/Diskussion - JavaScript

Zur Darstellung des Kommunikationsflusses zwischen Client und Server dürfen gerne ergänzende Diagramme (insbesondere UML Sequenzdiagramme) verwendet werden.

Pro Person 10 Minuten. Die Aufteilung zwischen Client und Server sollte ca. 50/50 sein; max 30/70.

Abgabe ist bereits eingerichtet (Deadline siehe Moodle!). Es kann ein Zip-Archiv und ggf. ein PDF hochgeladen werden.

Bewertungskriterien bei der JavaScript Präsentation

Es ist insbesondere darzustellen wie der Kommunikationsfluss zwischen Client und Server umgesetzt wurde. Hierbei ist insbesondere auf den Einsatz der verwendeten Technologien (z.B. WebSockets, Fetch API) einzugehen und wann welche Nachrichten ausgetauscht werden.

Weiterhin bietet es sich ggf. an die Strukturierung darzustellen und - falls vorhanden - wie die Cheatingprävention umgesetzt wurde.

27.10.: **Abgabe der finalen Lösungen (d.h. des finalen Codes) über Moodle**

Abgabe ist bereits eingerichtet (Deadline siehe Moodle!). Es kann nur ein Zip-Archiv hochgeladen werden.

29.10.: Spielzeit - wir spielen alle Spiele an dem Tag - ca. 10 Minuten pro Spiel; ggf. inkl. einer kurzen Einführung in das Spiel.

Stellen Sie sicher, dass Ihr Spiel in dieser Zeit vollständig spielbar ist!

Achtung!

Wenn Sie den Code aus Ihrem Editor heraus präsentieren, dann stellen Sie sicher, dass Ihr Theme einen guten Kontrast auf einem Projektor liefert und alles gut lesbar ist!

Einsatz von KI

Der Einsatz von KI-Tools (z.B. ChatGPT, GitHub Copilot) ist erlaubt, solange die Spiele den Anforderungen entsprechen und **vollständig eigenständig** präsentiert werden können!
