

WE1 2021 Testat

Zeitplan / Ablauf / Deadlines / Anforderungen

26.10.	Ausgabe der Testat-Aufgabe
14.11.	Testat für Peer-Review «Offline» abgegeben (Schritt 1)
21.11.	Peer-Review «Offline» abgeschlossen
05.12.	Testat für Peer-Review «Ajax» abgegeben (Schritt 2)
12.12.	Testat Peer-Review «Ajax» abgeschlossen
19.12.	Testat finale Version «Ajax» abgegeben

Durchführung

Das Testat ist eine Einzelarbeit.

Vorlage

Verwenden Sie die Vorlagen. Diese ist bereits richtig konfiguriert. Folder-Struktur und die API des Moduls *game-service* sollten nicht verändert werden.

Geben Sie auf der Webstorm console (oder Mac Terminal/Windows Console) im Wurzelverzeichnis des Projekts **npm install** ein um alle Abhängigkeiten zu installieren.

Folgende Befehle sind dann möglich

Befehl	Beschreibung
<code>npm run stylelint</code>	Testet ob CSS-Files in Ordnung sind.
<code>npm run w3c</code>	Testet ob HTML-Files in Ordnung sind.
<code>npm run eslint</code>	Testet ob JS-Files in Ordnung sind.
<code>npm run all</code>	Führt Tests für CSS/HTML/JS aus.

Wichtig: die Abhängigkeiten werden in den Folder *node_modules* abgelegt. Dieser Folder wird nicht abgegeben.

Getting Started

1. Entpacken Sie die Vorlage.
2. Installieren Sie die Abhängigkeiten (`npm install`).
3. Testen Sie, ob alles richtig installiert wurde
 - Console: «`npm run all`» im Root vom Projekt
 - Erwartete Ausgabe: 4 Warnungen («unexpected console state no-console» und «`npm run all completed`»)

Peer-Review

Teil des Testats sind 2 Peer-Reviews. Die Peer-Reviews werden in Moodle kommuniziert.

- Stellen Sie sicher, dass Ihre Implementation schon vor der Abgabe für die Peer-Reviews den Anforderungen der Checkliste entspricht.
- Die Reviews sind nicht verdeckt. Sie sehen für welche Person Sie den Review machen und in der von uns aktivierten nächsten Phase sehen Sie wer für Sie die Reviews gemacht hat.
- Reviews sollten ehrlich, nach bestem Wissen und Gewissen gemacht werden. Wir werden dies Stichprobenweise überprüfen.

- Beim Peer-Review sollten Sie Ihren zugeteilten «Peers» eine Aussage zur Checkliste machen und Hinweise geben wie die identifizierten Fehler reduziert bzw. eliminiert werden können. Das heisst es ist konstruktive Kritik gefragt.
- Die Durchführung aller zugeteilten Peer-Reviews in genügender Qualität ist Pflicht.
- Nicht durchgeführte oder offensichtlich falsche Peer-Reviews führen dazu, dass Ihr Testat als «nicht erfüllt» gewertet wird.
- Die aus dem Peer-Review des zweiten Schrittes resultierende Evaluation Ihrer Peers bildet den Ausgangspunkt unserer abschliessenden Bewertung. Zeigte schon die Bewertung Ihrer Peers eine ungenügende Leistung, und ist es diese auch noch bei Ihrer Endabgabe, dann ist diese Bewertung definitiv. Bei einer Bewertung durch uns als «nicht erfüllt» ist in diesem Fall keine Nachbesserung mehr möglich.

Finale Abgabe

Ihre korrigierte Version können Sie als Abgabe auf Moodle hochladen. Die Ablage wird rechtzeitig im Moodle aufgeschaltet und kommuniziert.

Die finale Abgabe muss folgende Kriterien erfüllen:

- Checkliste erfüllt
- Screenshot von «npm run all»
 - Keine (oder nur begründete Fehler/Probleme) im W3C Check
 - Keine (oder nur begründete Fehler/Probleme) im ESLint Check
 - Keine (oder nur begründete Fehler/Probleme) im StyleLint Check

Bestehens-Anforderungen

Für das Bestehen des Testats müssen folgende Bedingungen erfüllt sein

1. Sie haben beide Peer-Reviews abgegeben
2. Die Reviews sind inhaltlich korrekt (wird von uns in Stichproben überprüft)
3. Ihre finale Abgabe erfüllt die Checkliste

Nachkorrektur

Falls ihre finale Abgabe nicht in Ordnung ist, haben Sie die Möglichkeit einer Nachkorrektur. Diese Möglichkeit besteht nur, falls Ihr letzter Peer Review in Ordnung. Die Nachkorrektur wird von uns abschliessend bewertet (pass/fail). Eine zweite Nachkorrektur ist nicht möglich. Die Nachkorrektur wird rechtzeitig kommuniziert.

Aufgabenbeschreibung

Sie sollen das Spiel **Schere Stein Papier (Brunnen Streichholz)** implementieren. Das Testat ist in zwei Schritte aufgeteilt.

- Schritt 1: Offline Version
- Schritt 2: Online Version

Das auf Moodle publizierte Video gibt weitere Details zu einer *möglichen* Implementation der Funktionen. Das Aussehen und Layout Ihrer Abgabe muss nicht der im Video gezeigten Version entsprechen.

Wechsle zu Server

Michael! Wähle deine Hand!

Schere

= Stein

Papier

Brunnen

Streichholz

Nächste Runde in 1

Stein

1. Rang mit 13 Siegen

Michael

2. Rang mit 10 Siegen

Lisa

3. Rang mit 5 Siegen

Max

4. Rang mit 2 Siegen

Nadine

Zurück zum Ranking

Starten Sie ein neues Spiel

Name

Michael

Spiel Starten

History

Resultat	Spieler	Gegner
=	Stein	Stein
X	Streichholz	Stein
✓	Stein	Schere
✓	Papier	Brunnen

Schritt 1: Game-Local

Folgende Funktionen sind für Schritt 1 zu implementieren.

Startseite

- Lokale Rangliste darstellen
- Eingabe vom Namen
- Spiel starten, wenn Name eingegeben

Spielseite

- Spieler-Name darstellen
- Mögliche Hände darstellen
- Auswahl der Hand
- Wahl des Computers anzeigen
- Resultat anzeigen
- Zwischen den einzelnen Spielen ist eine Wartezeit einzuhalten (Optional in Schritt 1, Verpflichtend bei Schritt 2 und der Endabgabe)
- Die History zeigt alle Spiele der aktuellen Session an.
- Zurück zur Startseite

Testat Checkliste: Schritt 1

Folgende Checkliste bildet die Grundlage des ersten Peer-Reviews.

Kriterium	Evaluation
Kompletter Funktionsumfang ist vorhanden und funktioniert wie im Video: Startseite Rangliste, Eingabe vom Namen (Required), Spiel starten Spielseite Spieler-Name darstellen, Darstellen der möglichen Hände, Auswahl der Hand, Anzeigen der Computer-Wahl, Darstellen vom Resultat, History, Zurück zu Startseite.	Ja/Nein
W3C Check, Stylelint, ESLint ohne Fehler/Warnungen	Ja/Nein
Kein JavaScript im HTML-File	Ja/Nein
Game-Service wird genutzt (keine Netzwerk-Calls/fetch ausserhalb des Game-Service). Der Service ist unabhängig vom DOM, d.h. kein Zugriff auf DOM-Elemente im Service.	Ja/Nein

Folder-Struktur der Vorlage und API des game-service wurde beibehalten.	Ja/Nein
Webseite wird nie unresponsive (hängt) und erzeugt keine Fehler (Dev Konsole).	Ja/Nein
Alle Styles wurden über CSS-Regeln definiert. Es gibt keine Inline-Styles	Ja/Nein
Event-Bubbling wurde sinnvoll eingesetzt. Event-Handler werden nicht mehrfach registriert.	Ja/Nein
HTML-Elemente werden korrekt eingesetzt und nicht missbraucht. z.B. für Styling; <div> statt semantischen Tags	Ja/Nein
Code ist verständlich.	Ja / Nein (Kommentare notwendig)
Clean-Code Der JS/CSS/HTML-Code beinhaltet keine Probleme: <ul style="list-style-type: none"> • Keine globalen Variablen • Kein duplizierter Code • Sinnvolle und konsistente Namensgebung der Variablen / Funktionen / CSS-Klassen • Sinnvolle Wahl von HTML-Elementen (z.B. div statt ...). • Kein !important im CSS. • Kein «dead code» 	Ja / Nein (Kommentare notwendig)
Es wurden nur im Unterricht eingeführte Libraries eingesetzt. Zusätzlich erlaubt: momentjs	Ja/Nein
Abgabe ist eine Single Page App	Ja/Nein

Schritt 2: Game-Server

Im Schritt 2 soll Ihre App mit dem Game-Server kommunizieren. Der Game-Server wird folgendermassen angesprochen:

Method	URL	Beschreibung	Rückgabe
GET	https://stone.dev.ifs.hsr.ch/ranking	Daten für die Rangliste	Beispiel
GET	https://stone.dev.ifs.hsr.ch/play?playerName=DemoUser&playerHand=Stein	Lässt Server ein Spiel auswerten	Beispiel
GET	https://stone.dev.ifs.hsr.ch/play?playerName=DemoUser&playerHand=Stein&mode=normal	Lässt Server ein Spiel auswerten. System-Wahl ist auf die Werte Schere, Stein, Papier reduziert	Beispiel

Folgende Funktionen sind im Schritt 2 zu implementieren:

- Startseite
 - Wechseln zwischen lokalem und Server-Modus
 - Darstellen vom Ranking vom Server oder Lokal
- Spielseite
 - Das Spiel soll den korrekten Services nutzen (Server oder In-Memory).
 - Zwischen den Spielen ist eine Wartezeit einzuhalten.

Testat Checkliste: Schritt 2

Die folgende Checkliste ergänzt die Checkliste von Schritt 1 und bildet die Grundlage des zweiten Peer-Reviews.

Kriterium	Evaluation
<p>Kompletter Funktionsumfang ist vorhanden und funktioniert wie im Video:</p> <p>Startseite Wechseln des Modus («Lokal», «Server») Im Modus «Server» wird das vom Server gelieferte Ranking angezeigt, im Modus «Lokal» das im Browser verwaltete Ranking</p> <p>Spielseite Das Spiel kontaktiert im Modus «Server» den Server für die Spielauswertung. Im Modus «Lokal» findet die Auswertung im Browser statt.</p>	Ja/Nein
Für Server-Anfragen benutzt wird <code>fetch()</code> benutzt	Ja/Nein
Zwischen den Spielen muss gewartet werden und es ist keine Eingabe möglich. Dies wird vom System dargestellt.	Ja/Nein