



UE3 – JavaServer Faces (JSF) (25 Punkte)

Ziel dieses Übungsbeispiels ist die Erstellung einer JSF 2.0-basierten Web-Applikation, welche ein Memory-Spiel realisiert. Zusätzlich zur Funktionalität aus Übung 2 soll eine Registrierung und ein Login, sowie eine dynamische Spielfeldgröße (bis maximal 6x6) realisiert werden.

Deadline der Abgabe via TUWEL¹: **Mittwoch, 18. Mai 2011 23:55 Uhr**
Abgabegespräche: **Donnerstag, 19. Mai 2011**

Abgabemodalität

Beachten Sie die allgemeinen Abgabemodalitäten des TUWEL-Kurses¹. Zippen Sie Ihre Abgabe, sodass sie folgende Struktur aufweist:

- UE3-AG_<GruppenNr>_gruppe-<MatNr1>-<MatNr2>-<MatNr3>.zip
 - WEB-INF
 - ...(XML Config Files)
 - includes (Composition XHTML files)
 - Web Pages (Ordner)
 - ... (XHTML)
 - resources/css (CSS)
 - resources/img (Bilddateien)
 - resources/mycomponents (Custom Components)
 - src (Ordner)
 - ... (Quellcode)
 - ... (Internationalization)
 - UE3-AG_<GruppenNr>_gruppe-<MatNr1>-<MatNr2>-<MatNr3>.war

Angabe

Implementieren Sie eine JSF 2.0-basierte Web-Applikation, welche mit einer bereits eigens in UE2 entwickelnden Memory API ein webfähiges Memoryspiel realisiert. Als Entwicklungsumgebung (IDE) empfehlen wir NetBeans 6.9.1 und als Applicationserver GlassFish v3.0.1 (im entsprechendem NetBeans Bundle enthalten) oder Tomcat 7. Erstellen Sie ein neues Java Web Application Project. Verwenden Sie als User Interface den von Ihnen in der UE1 erstellten (oder in der Musterlösung zur Verfügung gestellten) XHTML- und CSS-Code – wo erforderlich können diese Dateien natürlich abgeändert werden. Die in UE2 erstellte API kann weiterverwendet und erweitert werden.

Anforderungen an Ihre Implementierung:

- Die Memory-API soll beliebig viele Karten-Paare unterstützen. Bei der Verwendung der API durch Ihre JSF Applikation sollen 2x2, 4x4 und 6x6 Karten unterstützt werden können.
- Es muss nur der Einspieler-Modus realisiert werden.

¹ <https://tuwel.tuwien.ac.at/course/view.php?id=2710>



- Es muss jedoch möglich sein, dass mehrere Spieler je ein eigenes Spiel im Einspieler-Modus am selben Server starten können. Diese dürfen sich nicht gegenseitig beeinflussen. Testen Sie dies mit Kollegen oder zwei unterschiedlichen Browsern. (Hinweis: Mehrere Tabs innerhalb eines Browsers benutzen die selbe Session)
- Die Spielinfo (gefundene/restliche Paare, Versuche, Zeit) muss dynamisch neben dem Spielbrett (siehe `table.html` in den Angaberessourcen) ausgegeben werden.
- Das User Interface muss natürlich soweit wie mit JSF 2.0 möglich den Anforderungen von XHTML 1.0 sowie WCAG-AA gerecht werden.
- Implementieren Sie eine Registrierung und ein Login. Es muss keine Datenbankbindung realisiert werden. Die Registrierung aller User muss jedoch für die Dauer der Applikation (bis zu einem neuen Deploy oder Restart des Servers) erhalten bleiben. Der Benutzer soll zu Beginn auf die Login Seite kommen. Falls er noch keine Login Daten besitzt, kann der User mittels Link auf die Registrierungsseite gelangen. Nach der Registrierung kommt er wieder auf die Login Seite. Bei Erfolgreichem Login kann der Spieler das Spiel beginnen.
- Alle Felder der Registrierung müssen validiert werden (nur Buchstaben in Loginnamen, Passwort muss mindestens 8 Zeichen lang sein und zumindest einen Buchstaben und eine Ziffer enthalten). Für das Geburtsdatum erstellen sie einen Custom Validator im Format TT.MM.JJJJ.
- Klickt der User bei der Registrierung auf die Checkbox „Zusätzliche Daten angeben“, so werden diesem die Felder für Persönlichen Daten angezeigt, ansonsten nicht.
- Realisieren Sie das Login mittels einer **Composit Component**.
- Verwenden Sie zur Erstellung der Seiten **Facelet Templates** und **Compositions**.
- Gestalten Sie das Spiel so, dass es sowohl in Englisch als auch in Deutsch angezeigt werden kann (Internationalization I18n)
- Beim Umdrehen einer Karte soll nicht die gesamte Seite neu geladen werden sondern nur der Spielbereich, der Spielstand und die Spielinformationen neu übertragen werden (AJAX component)

Angaberessourcen

Die Angaberessourcen können in TUWEL heruntergeladen werden. Diese beinhalten die folgenden Dateien:

- UE3-Angabe.zip
 - UE3.pdf (Dieses Dokument)
 - img (Ordner)
 - cards (Ordner)
 - ... (18 Bilder für 36 Karten-Vorderseiten)
 - ... (Sonstige Bilder, die in der Seite benötigt werden)
 - styles (Ordner)
 - screen.css (CSS-Datei der Musterlösung)
 - table.html (HTML-Datei der Musterlösung)



- o `register.html` (HTML-Datei der Musterlösung)

Hinweise

Das SurfStore Beispiel, welches Sie auch über TUWEL runterladen können, zeigt Ihnen die Verwendung von JSF 2.0. Es wird daher empfohlen, sich dieses Beispiel und den Source Code genau anzusehen. Zusätzlich zu den Folien der JSF Vorlesung empfehlen wir Ihnen, die BIG-internen Folien aus dem letzten Jahr (vor allem zusätzlich Templates [S 27], Compositions [S 28], Composit Components [S 29] und Internationalisierung I18N [S 46]) und das Beispiel JSF2.0 Sample Store durchzusehen. Wir sind uns im Klaren, dass einige dieser Features in der Vorlesung aus Zeitgründen nur angeschnitten wurden, sie aber dennoch einen wichtigen Teil von JSF 2.0 darstellen und mit den angegebenen Ressourcen leicht zu verwenden sind. Es wird Ihnen viel zur Lösung der Übung beitragen. Des Weiteren ist <http://jsfatwork.irian.at/> sehr zu empfehlen.