



## UE4 – Rich Client Technologies and Web Services (25 Punkte)

Ziel dieses Übungsbeispiels ist es, das Memory Spiel aus den Übungen 1-3 um Web Services sowie einen AJAX basierten Multiplayer Modus zu erweitern.

Deadline der Abgabe via TUWEL<sup>1</sup>: **Mittwoch, 15. Juni 2011 23:55 Uhr**  
Abgabegespräche: **Donnerstag, 16. Juni 2011**

### Abgabemodalität

Beachten Sie die allgemeinen Abgabemodalitäten des TUWEL-Kurses<sup>1</sup>.  
Zippen Sie Ihre Abgabe, sodass sie folgende Struktur aufweist:

- UE4-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)
  - UE4-AG\_<GruppenNr>\_gruppe-<MatNr1>-<MatNr2>-<MatNr3>.war

### Angabe

Erweitern Sie die Abgabe aus den Übungen 1–3 um folgende Funktionalität:

#### 1. Multiplayer Modus (6 Punkte)

Erweitern Sie das Memory Spiel aus Übung 3 so, dass zwei Spieler an einem gemeinsamen Spiel teilnehmen können. Dabei können beide Spieler abwechselnd Spielzüge tätigen. Wird ein Paar gefunden, darf ein weiterer Spielzug getätigt werden. Beide Spieler sollen sehen, welche Karten umgedreht werden. Es soll nur dann ein Spielzug möglich sein, wenn der Spieler gerade an der Reihe ist.

---

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



Hinweise zur Realisierung:

- Nach dem Login soll nicht direkt ein neues Spiel gestartet werden, sondern zuerst auf einen Gegner gewartet werden (davor kann der Spieler noch den Kontinent bzw. die Stacksize auswählen, mit welcher das Spiel gestartet werden soll). Wenn sich ein zweiter Spieler einloggt, wird das Spiel gestartet.
- Der Multiplayer Modus soll mit Hilfe von ICEfaces Ajax Push<sup>2</sup> realisiert werden. D.h., nachdem ein Spieler eine Karte umgedreht hat, sehen beide Spieler die aufgedeckte Karte. Spielstand und Spielinformationen sollen ebenfalls nach jedem Zug aktualisiert werden.
- Es ist nicht nötig, dass mehr als zwei Spieler gleichzeitig spielen können.
- Die Stacksize und der Kontinent für das Spiel werden immer von dem Spieler genommen, der als erster eingeloggt war.
- Netzwerkunterbrechungen bzw. Verlust der Verbindung durch Schließen des Browsers müssen nicht berücksichtigt werden.
- Während des Spieles soll beiden Spielern angezeigt werden, wer an der Reihe ist.
- Achtung: Inkludieren Sie die Bibliothek `icefaces-ace.jar` nicht in den Classpath Ihres Servlet Containers und verwenden Sie immer `<ice:form/>` statt `<h:form/>`. Andernfalls kommt es zu Problemen im Internet Explorer.

## 2. Integration eines HighScore-Services (6 Punkte)

Nach erfolgreichem Abschluss eines Spiels sollen die Spielergebnisse, für beide Spieler getrennt, automatisch auf einem Highscoreboard gepostet werden. Das Highscoreboard ist als Facebook Application Wall<sup>3</sup> realisiert. Verwenden Sie die RestFB Library<sup>4</sup> oder die Graph API von Facebook<sup>5</sup>, um einen neuen Highscoreeintrag auf die Application Wall zu posten. Für diesen Vorgang müssen Sie nicht auf Facebook registriert sein!

Für die Realisierung dieser Aufgabe stehen Ihnen zwei Artefakte zur Verfügung, die sie verwenden müssen:

- FacebookConnector.java
- Score.java

---

<sup>2</sup> <http://www.icefaces.org/main/ajax-java/ajaxpush.iface>

<sup>3</sup> <http://www.facebook.com/apps/application.php?id=182396081803634&v=wall>

<sup>4</sup> restfb-1.6.3.jar in den Angaberesourcen, <http://restfb.com/>

<sup>5</sup> <http://developers.facebook.com/docs/reference/api> (Facebook Account notwendig) bzw. PDFs im TUWEL wenn sie über keinen Facebookaccount verfügen, oder sich keinen anlegen wollen.



Die Highscore Einträge müssen in folgendem Format an die Facebook Wall gepostet werden:

Beispiel A: `Points: 50; Name: Max Muster`

Halten Sie dieses Format unbedingt ein, da ansonsten das Auslesen von bestehenden Highscoreeinträgen nicht mehr möglich ist. Die Methode `getFacebookPublicationString()` in `Score.java` liefert Ihnen den korrekten String.

Für den Zugriff auf die Facebook Wall über die Graph API sind zwei IDs notwendig:

- App ID: 182396081803634
- App secret: b78130a706117c79d717dd0baa936e0a

### **Implementierung**

1. Berechnen des Highscoreergebnis
2. Abholen aller Highscoreergebnisse von der Facebook Wall und Parsen der Strings wie in Beispiel A dargestellt. Achten Sie darauf, dass Walleinträge nicht notwendigerweise nur Messages sein müssen. Es können auch Bilder, Links u. ä. sowie ungültige Highscoreergebnisse (zB `Points 50 - Name: Max Muster`) vorhanden sein. Ihr Parser muss dies entsprechend berücksichtigen. Aus diesem Grund ist es auch wichtig, dass sie selbst nur gültige Einträge auf die Wall schreiben!
3. Anhand der von Facebook abgeholten Highscoreergebnisse den eigenen Rang ermitteln
4. Eigenes Highscoreergebnis posten
5. Dem Benutzer das eigene Highscoreergebnis anzeigen
6. Zusätzlich werden bei jedem Spiel auf der linken Seite des Spielfeldes die besten 10 Ergebnisse angezeigt. Das heißt, dass auch vor jedem neuen Spiel die aktuellste Highscoreliste von Facebook geholt werden muss.



### Berechnung des Highscores

- Erfassen Sie bei jedem Spiel die Zeit, die ein Spieler zum Lösen des Spiels gebraucht hat – diesen Wert brauchen Sie für die Berechnung des Spielergebnisses. Zeigen Sie dem Benutzer auch die aktuell verstrichene Zeit an.
- Das Spielergebnis  $x$  berechnet sich aus 300 minus der Anzahl von Sekunden  $s$ , die zum Lösen des Spiels gebraucht wurden, dividiert durch die Anzahl der gefundenen Paare  $p$ .  
D.h.

$$x = 300 - (s / p)$$

Wenn ein Spieler gar keine Paare gefunden hat, so liegt auch kein Spielergebnis vor. In diesem Fall soll kein Eintrag im Highscoreboard erfolgen.

- Sobald ein Memoryspiel beendet ist (ein Benutzer hat alle Kartenpaare erfolgreich aufgedeckt) werden die Spielergebnisse automatisch für beide Spieler getrennt im Highscoreboard gepostet. Sobald das Ergebnis gepostet wurde, wird dem Spieler sein Rang in der Highscoreliste angezeigt.
- Achten Sie insbesondere auf ein vernünftiges Exception Handling für den Fall, dass zum Beispiel das Service nicht verfügbar ist oder Werte in falschem Format auf der Wall gepostet wurden (in diesem Fall ignorieren sie beispielsweise ungültige Einträge)

### 3. Dynamische Bildauswahl mittels Flag Service (8 Punkte)



Ziel dieser Aufgabe ist es Daten dynamisch mittels eines Web Services abzurufen und in Ihre Anwendung einzubauen. Hierzu dient das Flag Service welches unter

<http://pephimon.big.tuwien.ac.at/FlagService/flagservice?wsdl>

erreichbar ist. Erweitern Sie Ihre Applikation um folgende drei Funktionen:

### (1) Kontinent Auswahl (2 Punkte)

Beim Neustart eines Spiels soll dem Benutzer eine Auswahlliste mit verschiedenen Kontinenten angezeigt werden. Die Kontinente für die Auswahlliste bekommen sie mit Hilfe der `getSupportedContinents` Operation vom Web Service retourniert.

Beim Neuanlegen eines Benutzers soll dieser auch die Möglichkeit haben, seinen Lieblingskontinent auszuwählen. Dieser Kontinent bleibt so lange gespeichert, bis der Benutzer mit Hilfe der Choice-Box beim Neustart eines Spieles die Auswahl ändert.



### Registrieren

Bitte füllen Sie das untenstehende Formular aus!

#### Login Daten und Größe des Stack

Benutzername	<input type="text"/>
Passwort	<input type="password"/>
Größe des Stack	<input type="text" value="2x2"/>
Kontinent	<input type="text" value="Europa"/>

☒ Persönliche Daten anzeigen

#### Persönliche Daten

Vorname	<input type="text"/>
Nachname	<input type="text"/>
Geburtsdatum	<input type="text"/>
Geschlecht	<input type="text" value="weiblich"/>

### (2) Spielfeldgrößenauswahl (2 Punkte)

Beim Neustart eines Spiels wird bereits jetzt dem Benutzer eine Auswahlliste mit verschiedenen Spielfeldgrößen angezeigt. Neu hinzu kommt nun, dass die Spielfeldgrößen vom Web Service bezogen werden.



Befüllen Sie das Auswahlfeld für die Spielfeldgröße mit jenen Werten, die Ihnen die Operation `getSupportedGameSize` zurück liefert.

Beim Neuanlegen eines Benutzers soll dieser auch die Möglichkeit haben, seine bevorzugte Spielfeldgröße auszuwählen. Diese Spielfeldgröße bleibt so lange gespeichert, bis der Benutzer mit Hilfe der Choice-Box beim Neustart eines Spieles die Auswahl ändert.

### **(3) Dynamische Anzeige von Flaggen als Karten für das Memoryspiel (4 Punkte)**

Basierend auf der Auswahl des Kontinents und der gewählten Spielfeldgröße holen Sie mit Hilfe der `getFlags` Operation Ländernamen und URLs zu den Flaggenbildern der Länder, welche Sie für Ihr Memoryspiel verwenden.

### **Angaberessourcen**

Die Angaberessourcen können in TUWEL ab 16. Mai 2011, 18 Uhr heruntergeladen werden. Diese beinhalten die folgenden Dateien:

- UE4-Angabe.zip
  - UE4.pdf (Dieses Dokument)
  - Facebook-Connector.zip (Java Klassen für den Facebook Connector)
- ICEfaces 2.0.1-bin
- ICEfaces 2.0.1-src

### **Hinweise**

Im Rahmen des Abgabegespräches werden Theoriefragen zur Abgabe gestellt, für die insgesamt 5 Punkte vergeben werden.