

## Übungsblatt 2: JavaScript

### Abgabe:

Dieses Übungsblatt ist einzeln oder in einer Gruppe zu lösen (wir empfehlen allerdings es allein zu bearbeiten). Die Lösung ist bis **Montag, den 23. Mai 2011, 12:00 Uhr s.t.** über UniWorx (<http://www.pst.ifi.lmu.de/uniworx>) abzugeben.

Benennen Sie die Dateien nach dem Schema <Übungsblatt>-<Aufgabe>.<extension>. Packen Sie alle Dateien in eine ZIP-Datei und laden Sie diese bei UniWorx hoch.

### Inhalt:

Ziel dieses Übungsblattes ist, die JavaScript-Programmierung kennen zu lernen und einen Einblick in die grundlegenden Komponenten von JavaScript zu bekommen. Da wir in der Übung mit WebGL arbeiten werden, sind Kenntnisse in JavaScript notwendig, um die Aufgaben auf den folgenden Übungsblättern bearbeiten zu können. Bitte nutzen Sie also diese Chance, sich in die Technologie einzuarbeiten.

### Hintergrund:

In diesem Abschnitt finden Sie in jedem Übungsblatt Hilfsmaterialien, Anleitungen und andere frei verfügbare Informationsquellen die Ihnen bei der Bearbeitung helfen können.

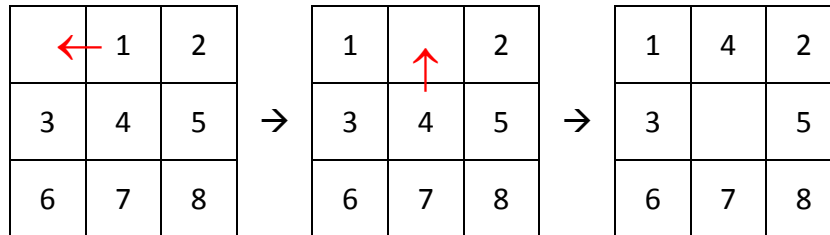
- <http://de.selfhtml.org/javascript>
- <http://www.w3schools.com/js/>
- [https://developer.mozilla.org/en/Canvas\\_tutorial](https://developer.mozilla.org/en/Canvas_tutorial)
- <http://getfirebug.com/javascript>

### Aufgabe 1: Einstieg JavaScript

Programmieren Sie mit Hilfe von JavaScript einen Taschenrechner. Verwenden Sie dazu Textfelder, Buttons und onclick-Events.

### Aufgabe 2: Schiebepuzzle

In dieser Aufgabe soll mit Hilfe von JavaScript ein Schiebepuzzle programmiert werden. Das Schiebepuzzle besteht aus einem 9 x 9 Gitter, in dem 8 Puzzleteile randomisiert angeordnet werden. Ziel des Spiels ist es, einzelne Teile des Puzzles so zu verschieben, dass das Bild wieder richtig zusammengesetzt wird. Dabei können nur Puzzleteile verschoben werden, die sich neben dem leeren Gitterfeld befinden. Folgende Abbildung zeigt ein Beispiel mit zwei aufeinanderfolgenden Verschiebungen:



Ihre Abgabe sollte folgende Anforderungen erfüllen:

- Es wird ein *Canvas* verwendet
- Es wird nur eine Bilddatei verwendet (d.h. es wird nicht für jedes Puzzleteil eine eigene Bilddatei erstellt)
- Es wird eine eigene (Javascript-)Klasse für die Spiellogik verwendet
- Verschiebung eines Puzzleteils erfolgt durch einen Mausklick auf das entsprechende Puzzleteil
- Der Nutzer erhält eine Benachrichtigung, wenn er das Puzzle gelöst hat.

### **Aufgabe 3 (\*): Document Object Model**

In dieser Aufgabe soll mit Hilfe von JavaScript und DOM eine kleine Visualisierung für eine Songliste erstellt werden. Die Daten stehen als XML zur Verfügung und enthalten die regulären Musikmetadaten, den Playcount und andere Informationen. Seien Sie kreativ 😊

Laden Sie sich für die Aufgabe die Dateien `music.xml` und `music.html` von der Vorlesungswebseite herunter. Die HTML-Datei enthält schon ein Codegerüst, das die XML-Daten aus `music.xml` einliest. (Alternativ können Sie auch direkt eine Musikliste von `last.fm` laden – die API für den Zugriff finden sie unter <http://www.last.fm/api>, wir haben für die Demodaten die Funktion `user.getTopTracks` verwendet. Ändern Sie den Parameter für den Benutzernamen im Beispiel-API-Aufruf für die Funktion in ihren eigenen Last.fm-Namen um mit Ihren Daten zu arbeiten).

Hinweis: Diese Aufgabe ist eine Stern (\*) - Aufgabe, d.h. wir präsentieren die beste Lösung auf der Vorlesungshomepage!