

# Projektseminar: Rechnergestützte Musikanalyse (WS 2005/06)

Leitung: Klaus Frieler

## Vorläufiger Terminplan

- 28.10.** Einführung & Besprechung
- 4.11.** Basiswissen Informationstechnologie (Übungszettel 1)
- 11.11.** Kodierung von Musik (Übungszettel 2)
- 18.11.** (3-stündig) Kurzvorstellung der Projekte
- 25.11.** fällt aus
- 2.12.** (3-stündig) Einführung in Javascript I (Übungszettel 3)  
(Entwicklungsumgebungen, HTML, Variablen, Datentypen)
- 9.12.** (3-stündig) Einführung in Javascript II (Übungszettel 4)  
(Kontrollstrukturen und Funktionen)
- 16.12.** (3-stündig) Diskussion der Projektansätze
- 23.12.** fällt aus
- 13.1** (3-stündig) Einführung in Javascript III (Übungszettel 5)  
(Objekte, die String- und Math-Objekte, reguläre Ausdrücke)
- 20.1** fällt aus
- 27.1** (3-stündig) Projektarbeit/berichte/diskussion
- 3.2** Projektarbeit/berichte/diskussion
- 10.2** Projektberichte, allgemeine Diskussion, Evaluation

## Scheinkriterium

- Gruppenprojekt mit schriftlichem Projektbericht
- erfolgreiche Bearbeitung der Übungszettel

## Arbeitsweise

Das Seminar ist zweigleisig angelegt: Neben einer Einführung in die Informationstechnologie und die Grundlagen der Programmierung anhand von Javascript sollen Projekte zur rechnergestützten Musikanalyse konzeptioniert und evtl. auch durchgeführt werden. Sowohl die Übungsaufgaben als auch die Projekte sollen von Gruppen aus 2-3 Mitgliedern bearbeitet werden. Der Schwerpunkt der Projektarbeit liegt dabei auf Entwicklung eines Projektkonzeptes. Das umfasst das Design von Algorithmen, und die Auswahl von Methoden und Softwaretools, die Beschaffung und Aufbereitung von Daten und die Entwicklung von Methoden zur Evaluation. Die Projektkonzepte sollen Grundlage einer Umsetzung dienen, die durch die Gruppenmitglieder oder den Seminarleiter erfolgen können, evtl. erst in einem Folgeseminar nächstes Semester. Über den Fortgang der Projektarbeit soll regelmäßig berichtet und im Plenum diskutiert werden. Am Ende

steht ein (gemeinschaftlicher) Bericht der Gruppen über den jeweiligen Projektstand, vorzugsweise einem fertigen Konzept.

**Übungsaufgaben** Es wird voraussichtlich 5 Übungszettel mit jeweils drei Aufgaben geben. Die Übungsaufgaben müssen von den einzelnen Gruppen jeweils zur nächsten Sitzung beim Seminarleiter abgegeben werden (schriftlich oder per Mail oder sonstwie elektronisch), die dann ausführlich besprochen werden. Jede Aufgabe wird mit Punkten bewertet. Zur Erlangung eines Scheines müssen mindesten 50 % der aller möglichen Punkte aller Aufgaben erreicht worden sein. Jedem Gruppenmitglied werden dabei alle Punkte eines gemeinschaftlich bearbeiteten Übungszettels gutgeschrieben.

## Projekte

Ein Wort vorweg: Nicht abschrecken lassen! Einige der hier folgend aufgelisteten Projekte sind so komplex und umfangreich, dass sie gar für Magister- und Doktorarbeiten taugen würden. Die Umsetzung der Projekte ist deshalb auch nicht primärer Sinn und Zweck dieser Veranstaltung, sondern die möglichst umfassende Konzeptionierung. Vor allem die eigenständige Einarbeitung in musikwissenschaftliche Themen und Probleme und die Entwicklung eigener Ansätze dazu, die auf dem Einsatz rechnergestützter Methoden beruhen, sollen im Vordergrund stehen. Die gedankliche Auseinandersetzung und die Gruppenarbeit bildet dabei den Schwerpunkt! Ein möglichst unvoreingenommenes Herangehen an die Aufgaben ist dabei ausdrücklich gewünscht. Desweiteren wird hier der problemorientierte Ansatz vertreten: Entwicklung oder Auswahl geeigneter Methoden für ein konkretes wissenschaftliches Problem, das es zu bearbeiten gilt.

## Harmonik

- **Harmonische Ähnlichkeit.** Entwicklung von Ähnlichkeitsmaßen für Akkordfolgen.
- **Implizite vs. explizite Tonalität.** Explorativer Vergleich von impliziter und expliziter Tonalität in Pop und Jazz-Stücken.
- **Automatische Harmonieanalyse.** Automatisierte funktionsharmonische Analyse.
- **Klassifikation von Rock/Pop- Akkordfolgen** Erstellung eines Katalogs von Pop/Rock-Akkordfolgen.

## Melodik

- **Virtuelle Polyphonie.** Extraktion virtueller Melodiestimmen aus Solowerken des Barocks.
- **Segmentierung.** Automatische Segmentierung einstimmiger Melodien.

- **Mozart oder Beethoven, Bach oder Händel?** Automatische Klassifikation und/oder Komponistenattribution klassischer und/oder barocker Komponisten (algorithmisch oder statistisch).
- **Westpolnisch oder Altdeutsch?** Automatische geographischen/stilistische Zuordnung von Folksongs (algorithmisch oder statistisch).
- **Katalog der Licks** Automatische Kategorisierung von Rock, Blues, Jazz- Licks.
- **Konturklassen** Automatische Klassifikation von Melodien nach Konturklassen.
- **'Variationator'** Design eines Algorithmus zur automatischen Generierung von Variationen zu einer gegebenen Melodie.

### Rhythmik

- **Rhythmische Komplexität.** Entwicklung von Maßen rhythmischer Komplexität einstimmiger Rhythmen.
- **Klassifikation von Drumpatterns** Automatische Klassifikation und stilistische Zuordnung von 1-2 taktigen Drumpatterns.

### Formanalyse

- **Automatische Formanalyse von Popsongs** Entwicklung von Verfahren zur Zerlegung eines (MIDI-kodierten) Popsongs in seine typischen Bestandteile (Intro, Strophe, Refrain, Bridge etc.)

## Javascript

Um die Grundlagen der Programmierung und des informatischen Denkens zu lernen wird eine Einführung in die Scriptsprache Javascript gegeben, zusammen mit praktischen Übungen, u.a. im neuen Rechnerpool des Institutes. Javascript wird zwar vor allem im Web-Bereich eingesetzt und selten zu anderen Aufgaben, schon gar nicht in der RMA, doch gibt es gute Gründe, Javascript zu lernen:

- Einfach zu erlernende Sprache mit allen wichtigen Elementen der gängigen Programmiersprachen.
- Hohe Verfügbarkeit. Jeder kann zu Hause Javascript programmieren ohne weitere Voraussetzungen. Jeder beliebiger Webbrowser dient dabei als Javascript-Interpreter. Ich empfehle den Einsatz von Firefox, wegen der besseren Debuggingausgaben als etwa Internet Explorer.
- Syntaktische Ähnlichkeit zu anderen Sprachen. Die grundlegenden Elemente und die Javascriptsyntax sind fast komplett identisch zu C/C++, Java, PHP etc.

- Objektorientierte Programmierung ist ansatzweise möglich.
- Javascript ist keine reine Lehrsprache, sondern wird professionell überall im Web eingesetzt.

Es gibt unzählige Bücher zu Javascript und HTML am Markt, aus denen ich keine Auswahl zu treffen vermochte. Wir werden SELFHTML als begleitende Literatur benutzen, dass sowohl gute Tutorials umfasst, als sich auch als ein hervorragendes und umfassendes Referenzwerk erwiesen hat. SELFHTML ist online verfügbar unter: <http://de.selfhtml.org/>