Übung 1

Einführung

Hinweise: Das JAVA Development Kit (JDK) zum Entwickeln von JAVA-Programmen und das JAVA Runtime Environment (JRE) zum Ausführen von JAVA-Programmen finden Sie für verschiedene Betriebssysteme unter
http://www.oracle.com/technetwork/java/javase/downloads/index.html
.

Bitte achten Sie darauf, dass die Version der installierten Java-Runtime und des Browser-Plugins genau der Version des installierten Java-Compilers entspricht!

Eine Entwicklungsumgebung für Java finden Sie z.B. unter
http://www.eclipse.org/

1. Machen Sie sich mit der Java-Umgebung vertraut (dies sollte eigentlich für Sie eine Wiederholung aus früheren Vorlesungen sein). Stimmen Programmsuchpfade, d.h. werden Java-Compiler (**`javac`**) und Runtime-Umgebung (**`java`**) gefunden (**`PATH`**-Variable)? Ist der **`CLASSPATH`** richtig gesetzt und enthält auch das aktuelle Verzeichnis „**`.`**“? Ist die gewünschte IDE (eclipse, J++, Java Editor...) funktionsfähig und kennt ebenfalls die genannten Programme und Suchpfade? Übersetzen Sie das allerseits bekannte Java „Hello World“-Programm und lassen Sie es in der Java-Runtime bzw. der IDE ablaufen.

   Das sollte auch ohne Musterlösung gehen.

2. Schreiben Sie in einer beliebigen Programmiersprache ein Programm, das folgende Beschreibung erfüllt[1]:

   > *„Ein Programm, das vollautomatisch bzw. mit wenigen Vorgaben neue Musikstücke in verschiedenen Stilrichtungen (Klassik, Pop, Metal, ...) komponiert und als **MP3-Datei** speichert.“*

   Als Anregung: Google hatte etwas ähnliches vor kurzem HTML-basiert → als Doodle. Ganz so graphisch ansprechend muss es natürlich nicht sein.

[1] Sie sollten nicht mehr als eine halbe Stunde für diesen Aufgabenteil investieren. Warum?

Die "unmögliche Aufgabe"

Wie Sie - hoffentlich - nach einer Recherche über Algorithmische Komposition erahnen können, ist dies trotz der „kurzen" Aufgabenstellung keine Aufgabe, die sich in ein paar Stunden erledigen lässt.

Themenbereiche, in denen Sie sich Expertenwissen aneignen müssten:

- *Künstliche Intelligenz,*
- *Psychoakustik,*
- *Generative Grammatik,*
- *evtl. Markow-Ketten,*
- *...*

Selbst dann, wenn Sie vorhandene Systeme mitverwenden, dürfte sich der Aufwand als wesentlich höher herausstellen, als es die scheinbar sehr knappe Aufgabenstellung vermuten lässt. Sinn dieser Übung war es, zu erkennen, ***dass die Aufgabenstellung unmöglich alleine im Rahmen einer Übung zu lösen ist****. ;−)*

Diese Aufgabe, die einen einzelnen Entwickler klar überfordern würde, ist es, soll zu einer ersten Einschätzung für den Aufwand zu führen, den eine komplexe Software mit einer wenig präzisen Beschreibug verursachen kann. Bei dieser Aufgabe dürfte der Bedarf bei einer Neuentwicklung eines entsprechenden Programme bei mindestens 60 Personentagen liegen.

Unter Mitverwendung von Open Source Komponenten wie SoundHelix (JAVA) oder OpenMusic (Lisp) kann man immerhin schnell einen Prototypen (kommt später in der Vorlesung noch) generieren, der auch als Basis des geforderten Programms dienen kann. Allerdings nur, sofern dies durch die Mitverwendung von GPL-lizensierten Komponenten auch unter einer entsprechenden Open Source Lizenz stehen darf.

3. Führen Sie eine Anforderungsanalyse und Aufwandsabschätzung für die in der vorigen Aufgabe spezifizierte Software durch, mit Ansätzen in Form eines vorläufigen Lastenhefts.

   (a) *Ausgangssituation und Zielsetzung: Erstellung eines Musik komponierenden Programms.*

   (b) *Produkteinsatz (Beispiel): Generierung von „angenehm hörbaren" Musikstücken durch einen Algorithmus, Ausgabeformat vorzugsweise MP3.*

   (c) *Produktübersicht: Das Programm soll nach Vorgaben einer oder mehrere Kategorien und Parametern (Tempo, Rhythmus, Stil) eine entsprechende musikalische Komposition durchführen und speichern. Der Benutzer kann die Eingabe der Parameter per Kommandozeilenaufruf, oder dialog/menügesteuert dem Programm übergeben.*

   (d) *Funktionale Anforderungen: Aus den getätigten Eingaben generiert der Algorithmus einen Vorschlag für ein Musikstück, in Form von aneinandergehängten Samples, die der Benutzer überprüfen und akzeptieren oder verwerfen kann. Die Ausgabe erfolgt über Lautsprecher oder Kopfhörer. An Systemvorraussetzungen soll*

eine Standard PC-Hardware „mittlerer Leistungsklasse“ zum Einsatz kommen, Betriebssystem GNU/Linux sowie eine aktuelle Glibc-Runtime sind vorinstalliert.

(e) *Nichtfunktionale Anforderungen*

i. *Benutzbarkeit: Die Software soll auch von Laien bzw. Computer-Anfängern eingesetzt werden können, und darf nur ein Minimum an technischem Hintergrundwissen voraussetzen.*

ii. *Zuverlässigkeit: (Siehe auch Abnahmekriterien): Von drei automatisch generierten Musikstücken/Samples sollen mindestens 2 als „allgemein brauchbar“ von einem neutralen Zuhärer bewertet werden.*

iii. *Effizienz: Eine Lauffähigkeit auch auf alter Hardware ist wünschenswert, aktuelle kostengünstige Hardware ist zu unterstützen.*

iv. *Änderbarkeit: Spätere Erweiterungen wie Plugins für verschiedene Instrumentarien und Soundeffekte sind in der Schnittstellendefinition zu berücksichtigen, so dass das Programm später entsprechend weiterenwickelt werden kann.*

v. *Übertragbarkeit: Die Software soll sich leicht in bestehende Desktop-Systeme als Multimedia-Anwendung integrieren lassen, und durch eine Open Source Lizenz in ihrer Weiterverwendung nicht eingeschränkt sein.*

vi. *Risikoakzeptanz: Die Software wird nicht in kritischen Bereichen eingesetzt, und kann daher früh produktiv eingesetzt und ggf. noch bis Projektende verbessert werden.*

(f) *Skizze der Systemarchitektur: Hatten wir noch nicht in der Vorlesung, aber so könnte es aussehen: (s. Tafel)*

(g) *Lieferumfang: Unter Ubuntu (TM) GNU/Linux installierbares Softwarepaket, technische Anleitung und kurze Benutzereinführung.*

(h) *Abnahmekriterien: Von 12 generierten Musikstücken sollen 2/3 als „anhörbar“ von einem neutralen Testhörer beurteilt werden.*

4. Richten Sie sich die ROBOCODE-Entwicklungsumgebung ein. Anleitung und Links hierfür finden Sie unter http://knopper.net/bw/sep/.

   Sollte so aussehen: