gmut

Das GUIDO-MUTABOR

26. Januar 1999

Inhaltsverzeichnis

Einleitung

`gmut` ist eine Kommandozeilenversion von MUTABOR II.WIN, welche als Ein- und Ausgabe Dateien im GUIDO-Format verwendet. Dieses Format wurde an der TU Darmstadt im Rahmen des Salieri-Projekts entwickelt. Das zentrale Ziel dabei sit die adäquate Repräsentation von Musik. Dabei ist es möglicht in die GUIDO-Datei Steueranweisungen für Mutabor mit Hilfe eines speziellen Tags `\mutabor` einzubauen. Die Mikrotonalen Änderunge, die MUTABOR erzeugt, werden in der resultierenden Ausgabe durch die Verwendung des `\alter`-Tags erreicht.

1 Aufruf von `gmut`

1.1 Kommandozeile

Das Programm kann mit auf der Kommandozeile (oder Eingabeaufforderung) in folgender Form aufgerufen werden:

gmut *Mutabor-Datei Routen* [*Starttasten*] [*Optionen*]

Mutabor-Datei ist eine ganz normale MUTABOR-Datei (siehe dazu die Beschreibung von MUTABOR). Es ist lediglich zu beachten, dass `gmut` als Datei-Erweiterung `.mut` (statt `.mus`) erwartet und gegebenfalls anfügt.

Routen ist eine Spezifikation, wie die Musikdaten im GUIDO-Format verarbeitet werden sollen, woher sie kommen und wohin sie gespeichert werden sollen.

Bei Version 0.60 beschränkt sich die Angabe der Routen auf eine `.gmn`-Musikdatei im GUIDO-Format. Diese kann auch zur Steuerung von `gmut` das Tag `\mutabor` enthalten, welches in Abschnitt 2 beschrieben ist. Als Datei-Erweiterung wird gegebenfalls `.gmn` angefügt. Die Ausgabe wird in eine Datei mit demselben Namen aber mit der (leicht geänderten) Erweiterung `.gmm` geschrieben.

Ab Version 0.70 können auch mehrere Routen durch Leerzeichen getrennt angegeben werden, um die Datenströme aufzufächern und zu verschmelzen. Wie die Routen aufgebaut sind, finden Sie im folgenden Abschnitt 1.2.

Starttasten ist eine Sequenz von Buchstaben die als Initialisierung für das Instrument bezihungsweise die Box Nummer 1 verwendet wird. `gmut` reagiert so, als würden bei MUTABOR diese Zeichen auf der Tastatur eingegeben werden und stellt dementsprechend die Logiken und Tonsysteme ein. Wenn eine Logik und ein Tonsystem auf die gleiche Taste reagieren, wird die Logik bevorzugt. Um die Verwendung des „gleichbuchstabigen" Tonsystems zu erreichen, ist dem Buchstaben ein & voranzustellen (z. B. „`g&s`" bewirkt bei `demo.mut`: Logik „Gleich", Tonsystem „Sechzehntel"). Die Zeichen können mit und ohne Zwischenraum geschreiben werden.

Optionen sind mögliche Optionsschalter:

`/key:`*Nr*, *Text* ermöglicht es, an einzelne Boxen Tastatur-Zeichenfolgen zu übergeben, um Logiken und Tonsysteme zu aktivieren.

Nr ist dabei die Nummer der Box.

Text ist eine Zeichenfolge, die so wie sie ist für die Einstellung der Boxen verwendet wird. `gmut` reagiert so, als würden bei MUTABOR diese Zeichen auf der Tastatur eingegeben werden und stellt dementsprechend die Logiken und Tonsysteme ein. Wenn eine Logik und ein Tonsystem auf die gleiche Taste reagieren, wird die Logik bevorzugt. Um die Verwendung des „gleichbuchstabigen" Tonsystems zu erreichen, ist dem Buchstaben/Zeichen ein & voranzustellen (zum Beispiel „g&s" bewirkt bei demo.mut: Logik Gleich, Tonsystem Sechzehntel). Alle Tastenanschläge müssen dabei ohne Leerzeichen geschrieben werden.

`/silent` unterdrückt jegliche Ausgabe von Protokolldaten (dieser Modus ist bei Einbindung von `gmut` in andere Programme zu empfehlen)

`/report` zeichnet alle Protokolldaten in der Datei `GMut.rep` auf

`/help` gibt einen kurzen Hilfebildschirm aus

1.2 Routen ab `gmut` Version 0.70

Jede der Routen auf der Kommandozeile (siehe Abschnitt 1.1) ist folgendermaßen aufgebaut:

`Eingabedatei``(`*`Typ`*`,`*`von`*`,`*`bis`*`,`*`Box`*`)`*`Ausgabedatei`*

Dabei darf kein Leerzeichen verwendet werden. Die einzelnen Bestandteile sind:

`Eingabedatei` ist der Name einer Musikdatei im GUIDO-Format. Diese kann auch zur Steuerung von `gmut` das Tag `\mutabor` enthalten, welches in Abschnitt 2 beschrieben ist. Als Datei-Erweiterung wird gegebenfalls `.gmn` angefügt. Eine Eingabedatei kann auch bei anderen Routen wiederverwendet werden.

`Typ` gibt den Typ der Route an und filtert in Verbindung mit den Zahlen *`von`* und *`bis`* die Eingabedatei. Es gibt vier verschiedene Typen:

- `all` Alle Daten der Datei werden weitergeleitet, die Parameter *`von`* und *`bis`* sind wirkungslos.
- `staff` Es werden nur die Systeme von *`von`* und *`bis`* weitergeleitet. Die Nummern entsprechen der Nummer der Sequenz im Hauptsegment (also natürliche Zählung, das `\staff`-Tag wird nicht berücksichtigt).
- `box` Es werden die Sequenzen weitergeleitet, deren über das `\mutabor`-Tag zugewiesene Box-Nummer im Bereich von *`von`* bis *`bis`* liegt.
- `else` Es werden die Daten der Eingabedatei weitergeleitet, die nicht schon von einer anderen Route bearbeitet wurden (die Reihenfolge der Routen in der Kommandozeile ist entscheidend). Die Parameter *`von`* und *`bis`* sind wirkungslos.

Wenn *`Typ`* weggelassen wird, wird als Typ all verwendet.

`von` **und** *`bis`* geben an welche Systeme (Typ staff) oder Boxen (Typ box) verwendet werden, und zwar als Bereich von *`von`* bis einschließlich *`bis`*. Wenn *`bis`* weggelassen wird, wird der Wert von *`von`* eingesetzt (also genau ein System, eine Box), wenn auch*`von`* fehlt, wird für beide Werte 0 angenommen.

`Box` gibt an, in welcher MUTABOR-Box (welches MUTABOR-instrument) diese Daten verarbeitet werden sollen (entspricht der Nummer eines der 16 Instrumente in MUTABOR II). Fehlt dieser Wert, wird alles in die Box Nummer 0 geleitet. Wenn der Wert -1 ist, dann werden die Daten in die Boxen entsprechend der `\mutabor`-Tags, oder in die Box 0 geleitet, wenn keine solchen Tags vorhanden sind.

Ausgabedatei ist der Name der GUIDO-Datei, in die die mutierten Daten geschrieben werden sollen (falls bereits eine solche Datei existiert, wird diese ohne Rückfrage überschrieben). Als Dateierweiterung wird gegebenfalls `.gmn` angefügt. Wenn keine *Ausgabedatei* angegeben ist, werden die Daten nur verarbeitet (z. B. in der Harmonieanalyse) aber nicht ausgegeben. Die *Ausgabedatei* kann auch in weiteren Routen verwendet werden. Dann werden die Daten gemischt in die Datei geschrieben.

1.3 Beispiele

```
gmut demo.mut rondo.gmn(all,0,0,1)rondo_mu.gmn /key:1,n
```

erzeugt aus der Datei `rondo.gmn` die Datei `rondo_mu.gmn` unter Verwendung der MUTABOR-Datei `demo.mut`. Das Mutieren geschieht in der Box 1, welcher zu Anfang „n" übermittelt wird, d. h. es wird das tonale Netz eingestellt (Taste N).

```
gmut demo rondo(1)rondo_mu /key:1,"n"
```

tut genau dasselbe, hier wurden lediglich die Möglichkeiten des Weglassens von Parametern in Anspruch genommen.

```
  gmut demo noten1(all,0,0,1)noten noten2(staff,1,1,1)noten
/key:1,g&s
```

verschmilzt die Daten von `noten1.gmn` und dem ersten Notensystem von `noten2.gmn` zu einer Datei `noten.gmn`, wobei die Daten entsprechend der Datei `demo.mut` mutiert werden. `gmut` wird dazu am Anfang die Zeichenfolge „`g&s`" an MUTABOR übergeben, d. h. bei `demo.mut` würde in der Logik „Gleich" das Tonsystem „Sechzehntel" aktiviert.

```
gmut demo rondo(staff,1,0)rondo1st rondo(else,0) /key:1,"n
```

mutiert die Daten von `rondo.gmn` im Tonalen Netz. Es wird aber nur das erste System ausgegeben, und zwar in die Datei `rondo1st.gmn`.

```
gmut demo song(-1)song_mut /silent
```

mutiert die Daten von `song.gmn` und gibt sie als `song_mut.gmn` wieder aus. Ob die Daten dabei tatsächlich mutiert werden und in welchen Boxen, hängt davon ab, ob in `song.gmn` über das `\mutabor`-Tag Steuerinformationen enthalten sind. Dem Programm wird ein Maulkorb verpasst, das heißt es werden keine Information über den Verarbeitungsfortschritt ausgegeben.

```
  gmut my_phyta.mus song(box,3,5,-1)song_mut
trumpet(all,3)song_mut /key:3,x /key:4,u
```

verschmilzt die Sequenzen von `song.gmn`, für die über `\mutabor<"box", ...>` die Boxen 3, 4 oder 5 zugewiesen wurden und die Daten von `trumpet.gmn`. Die MUTABOR-Datei ist `my_phyta.mus`; die Daten von `song.gmn`, werden dementsprechend in den Boxen 3 bis 5 verarbeitet, `trumpet.gmn` wird in Box 3 geleitet. Box 3 wird am Anfang „`x`“ übergeben und Box 4 „`u`“.

```
gmut demo alle n
```

erzeugt mit `gmut` Version 0.60 (!) die Datei `alle.gmm`, die die Daten der Datei alle.gmn enthält, die jedoch dem tonalen Netz entsprechend (Taste N) mutiert sind.

2 Das Mutabor-Tag

Zur Steuerung von `gmut` kann in die GUIDO-Datei Steuerinformation in Form des `\mutabor`-Tags eingefügt werden.

Allgemein besitzt es die Form

```
\mutabor<"Parametername", spezifische Parameter ...>
```

Es sind folgende MUTABOR-Parameter implementiert:

Parameter-name	*Spezifische Parameter*	*Beschreibung*
`key`	String: Buchstabenfolge	Die Zeichenkette wird wie die Eingabe über Computertastatur bei MUTABOR zur Aktivierung der Logiken und Tonsysteme verwendet. Die Daten werden an das Instrument bzw. die Box übergeben, das gerade die Stimme verarbeitet, in der dieses `\mutabor`-Tag auftaucht.
`instrument`	Integer: Instrumentnummer	Dieses Tag gibt an, von welchem Instrument bzw. von welcher Box diese Stimme verarbeitet werden soll.
`box`	Integer: Instrumentnummer	wie Parameter „`instrument`“

Wenn `\mutabor` als ranged tag verwendet wird, wird es genauso verarbeitet, als wäre es ein Tag ohne range.

2.1 Beispiele

```
\mutabor<"key","gd">
\mutabor<"key", "h&s">
\mutabor<"instrument", 3>
\mutabor<"box", 3>
```

3 GMN2MIDI

`gmn2midi.exe` ist ein kleiner fragmentartiger GUIDO-zu-MIDI-Konverter, der in der Lage ist, das `\alter`-Tag zu verarbeiten, also Mikrotonalität ermöglicht. Dieses DOS-Programm ist aber kein vollständiger GMN-zu-MIDI-Konverter. Ein solcher wird aber hoffentlich bald zur Verfügung stehen. L. Martin von der GUIDO-Gruppe hat sich schon eine Weile damit befasst. GMN2MIDI ist somit nur als vorläufiges Behelfs-Tool zu sehen (oder hören), mit dem man die Wirkung von `gmut` auf spartanische Weise hörbar machen kann.

Um GMN2MIDI herunter zu laden klicken sie hier .

3.1 Beschreibung

Der Aufruf erfolgt über

> gmn2midi *gmn-Datei* [.gmn] [*MIDI-Datei* [.mid]]

gmn-Datei ist dabei eine ganz normale GUIDO-Datei

MIDI-Datei ist der Name der MIDI-Datei, die erzeugt werden soll.

3.2 Wichtige Hinweise zum Abspielen der erzeugten MIDI-Dateien

Nicht mit jedem MIDI-Fähigen Instrument werden Sie dieselbe Freude haben. Um Ihnen dennoch die besten Ergebnisse (andere sind meist inakzeptabel bis grausam) zu liefern, sollten Sie bei der Einstellung des Ausgabe-Gerätes auf folgendes achten:

- Das Ausgabegerät muss Pitch-Bending erlauben (Töne nach oben und unten „ziehen“ können).
- Der Pitch-Bend-Range (die Maximale Pitch-Weite) muß auf 2 Halbtöne eingestellt sein.
- Das Ausgabegerät muß sich im Multi-Mode befinden. Das heißt, das Gerät muss alle 16 Kanäle über MIDI ansteuern lassen und jedem Kanal seine eigenen Parameter (Programm/Instrument, Pitch-Bend usw.) zuordnen lassen. Beim Yamaha SY55 heißt der entsprechende Modus tatsächlich „Multi“, während beim Korg M1[1] eine entsprechende Ansteuerung im Sequencer-Modus („Seq“) möglich ist.
- Es werden unter Umständen alle 16 Midikanäle verwendet. (Also z. B. General-MIDI ausschalten.)

Es lohnt sich also, die Bedienungsanleitung des Gerätes zu lesen. Interne Soundsynthesen von Soundkarten entsprechen in der Regel diesen Anforderungen.

Na dann viel Spaß, und ... Ohren festhalten!

[1] Der Korg M1 unterstützt tatsächlich nur maximal 8 Kanäle.