



Musik und Komponieren mit Standardprogrammen

Erich Neuwirth

Universität Wien
Fachdidaktisches Zentrum für Informatik



Musik und andere akustischen Phänomene am Computer

Erich Neuwirth

Universität Wien
Fachdidaktisches Zentrum für Informatik

Musik als Transportmittel für Informatikkonzepte

Musik (und Sound) ist für viele Schüler interessant

Musikalische Ideen kann man relativ einfach diskutieren

Implementation ist mit verschiedenen Hilfsmitteln möglich



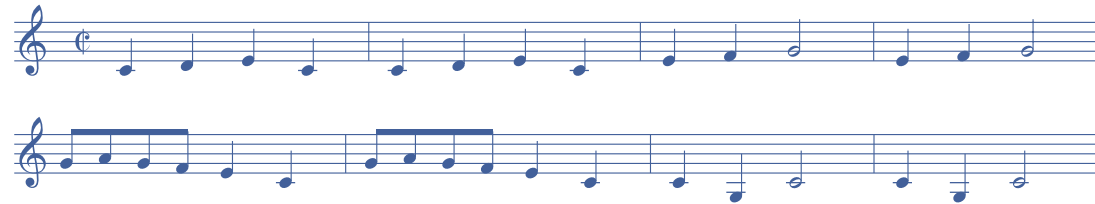
Wiedergabe von Musik



Repräsentationen von Musik

Frere Jacques

Trad.



X:1

T: Frere Jacques

C: Trad.

M: C|

L: 1/4

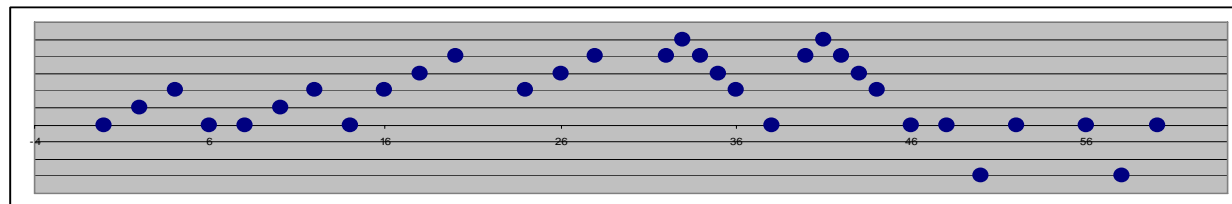
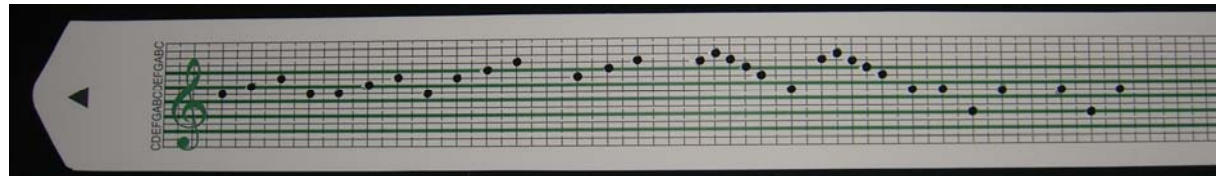
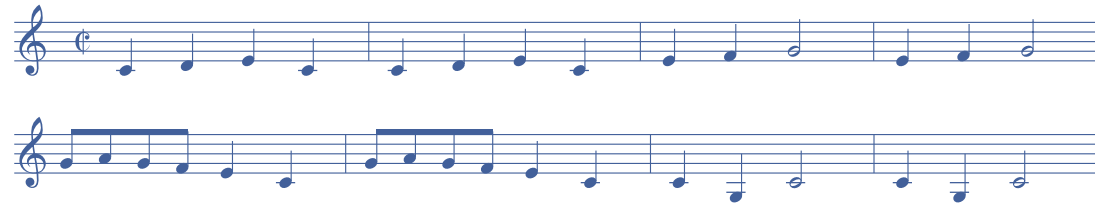
K:C

C D E C | C D E C | E F G2 | E F G2 |
G/2A/2G/2F/2 E C | G/2A/2G/2F/2 E C |
C G, C2 | C G, C2 |]

Repräsentationen von Musik

Frere Jacques

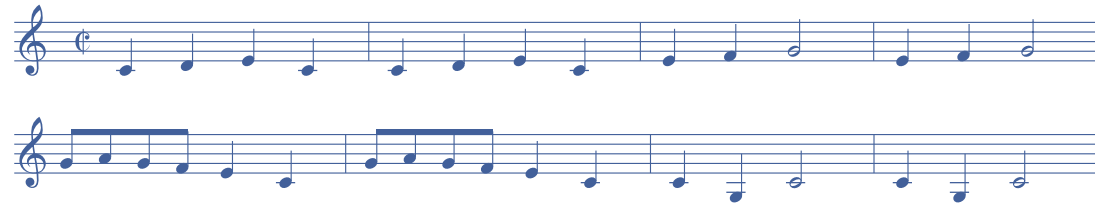
Trad.



Repräsentationen von Musik

Frere Jacques

Trad.



100	note	1	60	1	250
250	note	1	62	1	250
250	note	1	64	1	250
250	note	1	60	1	250
250	note	1	60	1	250
250	note	1	62	1	250
250	note	1	64	1	250
250	note	1	60	1	250

```
[ 0 [note 1 60 1 250]]
[250 [note 1 62 1 250]]
[250 [note 1 64 1 250]]
[250 [note 1 60 1 250]]
[250 [note 1 60 1 250]]
[250 [note 1 62 1 250]]
[250 [note 1 64 1 250]]
[250 [note 1 60 1 250]]
```

Haptische Manipulation in Tabellenkalkulation

Direkte Manipulation (Angreifen)

Strukturieren mit Programmblöcken

2-stufige Programmiersprache

Phrasensprache

Assembler, der auf Phrasen operiert

Phrase

frere	reltime				
100	note	1	60	1	250
250	note	1	62	1	250
250	note	1	64	1	250
250	note	1	60	1	250
250	note	1	60	1	250
250	note	1	62	1	250
250	note	1	64	1	250
250	note	1	60	1	250

Assembler

	clearall		
init	define	\$A\$20:\$D\$24	
frere	define	\$A\$1:\$E\$9	
frere1	copy	frere	
frere2	copy	frere	2
frere2	rechannel		1
frere2	timeshift		2000
all	copy	init	
all	merge	frere1	
all	merge	frere2	
all	play		

Erweiterung musikalischer Ideen

Position im Raum (Lautstärke, Stereoposition, Hall)

Projekt Geigerzähler: Welche Zufallszahlen stimmen?

Separate Ereignisströme: Prozesssynchronisation, Parallelisierung

Wenn die Projekte zu komplex werden,
unterstützt Tabellenkalkulation den Strukturierungsprozess
nicht gut genug



Höhere Programmiersprache: LOGO

Musik-Bibliothek

Einfache Programmierung:

Sequentialisierung von MIDI-Kommandos
und musikalischer Repräsentation

```
foreach [60 64 67] [noteon 1 ? 1  
                    waitmilli 250  
                    noteoff 1 ? 0]
```

Musikalische Operationen sehr unbequem

Andere Repräsentation:

```
to ourmel ody  
  output [[0 [note 1 60 1 500]] [500 [note 1 62 1 500]]  
         [500 [note 1 64 1 500]] [500 [note 1 67 1 500]]  
         [500 [note 1 72 1 250]] [250 [note 1 67 1 250]]  
         [250 [note 1 64 1 250]] [250 [note 1 62 1 250]]  
         [250 [note 1 60 1 1000]]]  
end  
playstream ourmel ody
```

Programmierkonzepte

Operationen auf Noten und Phrasen führen ganz natürlich zu Konstrukten höherer Ordnung (lambda expressions)

Syntaktisch einfache Sprache, also sind die Projekte "unterrichtstauglich"

Musik und akustische Phänomäne sind natürliche Beispiele von simultan ablaufenden Prozessen mit Synchronisation

Noten und Phrasen sind Datenstrukturen

Musikalische (kompositorische) Ideen müssen als Informatikkonzepte und Programmierkonstrukte entworfen und implementiert werden

Music kann auch als MIDI-Datei gespeichert werden:
Interpreter und Compiler





Typische Projekte

Einfache Melodie

Kanon

Geigerzähler

Marschierender Trommler

Marschierende Musikkapelle

Vorbeifahrendes Rettungsauto (Bewegung und Doppler-Effekt)

Project in Kooperation mit Northeastern University, Boston

Materialien

MidiCSD

<http://sunsite.univie.ac.at/musicfun/>

FMSLogo

<http://fmslogo.sourceforge.net/>

ABC Music Project

<http://abc.sourceforge.net/>

LilyPond (Notensatzsystem, Open Source, frei, anspruchsvoll)

<http://lilypond.org/>

Musette (freies Notensatzsystem, einfach)

<http://musettemusic.com/>

Finale Notepad (einfach, nicht frei, aber billig)

<http://www.finalemusic.com>