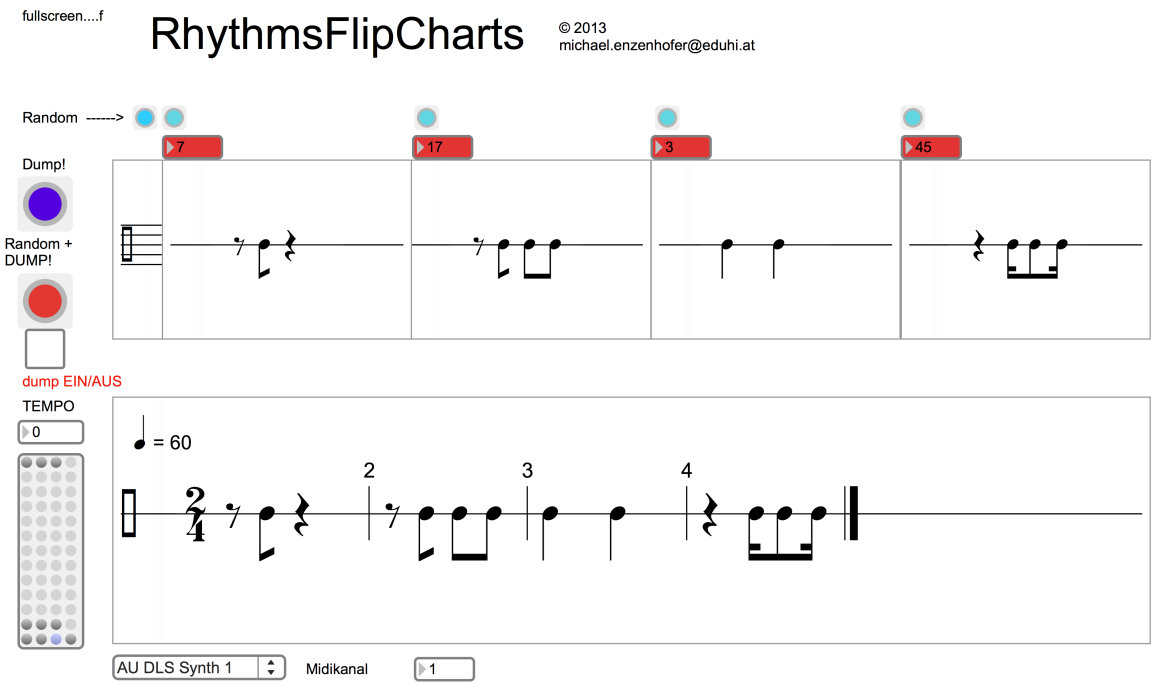
Abschließend ein kleines Beispiel mit *bach* in MAX erstellt:
RHYTHMSFLIPCHARTS setzt sich aus vier 2/4-Takten zusammen. Jeder Takt beinhaltet 50 idente mögliche Rhythmen, die per Zufall ausgewählt werden. Das ergibt pro „Knopfdruck" eine von 50 $x$ 50 $x$ 50 $x$ 50 = 6 250 000 – man kann es fast nicht glauben – möglichen 4-Takt-Gruppen. Für den Benützer (*User*) dieses MAX-*Patches* wurde der *Präsentationsmodus* gewählt, das heißt, man sieht nur die *UserInterface*-OBJECTS, die für die Verwendung notwendig sind. Die Programmierung mit den *bach*-OBJECTS und ihren Verbindungen und Vernetzungen bleiben verborgen.




## 4.2 bach-slots

Viele wichtige *bach*-OBJECTS und Einstellungsmöglichkeiten in *bach* sind bisher noch nicht behandelt. Der Einfachheit halber wurde nicht ausgeführt, dass neben den *inlets* für *Measureinfo, Cents* und *Durations* natürlich auch ein *inlet*

für *Velocities* vorhanden ist.

Steuerungsmöglichkeiten, wie sie von üblichen *Sequenzern* geläufig sind – etwa Abspielen, Pausieren, Tempoveränderungen, Loops und so weiter – sind Selbstverständlichkeiten.

In den Notensystemen können auch *Marker* gesetzt werden, um Positionen zielsicher und zeitlich exakt anzuspielen. Den einzelnen Notensystemen werden Midi-Kanäle zugeordnet, die – wie sonst auch üblich – die Klänge innerhalb des Computers oder externe Klangerzeuger anspielen.

Eine Besonderheit für Liebhaber der Komplexität stellt das *Extras-inlet*, welches auch die Artikulations-Angaben aufnimmt, dar. Dieses *inlet* versteht auch sogenannte *slot*-Informationen. Bis zu 35 *slots* können jeder einzelnen Note zugewiesen werden. Diese *slot*-Informationen bewirken einerseits die Darstellung beliebiger Klangveränderungen im Zeitverlauf – beispielsweise *Hüllkurven* – und andererseits die Ausgabe der zugewiesenen Parameter beim Abspielen der Notation.

Jede Note fasst also bis zu 35 *slots*. Wobei jedem *slot* eine von 16 Kategorien (*types*) zugewiesen werden kann.

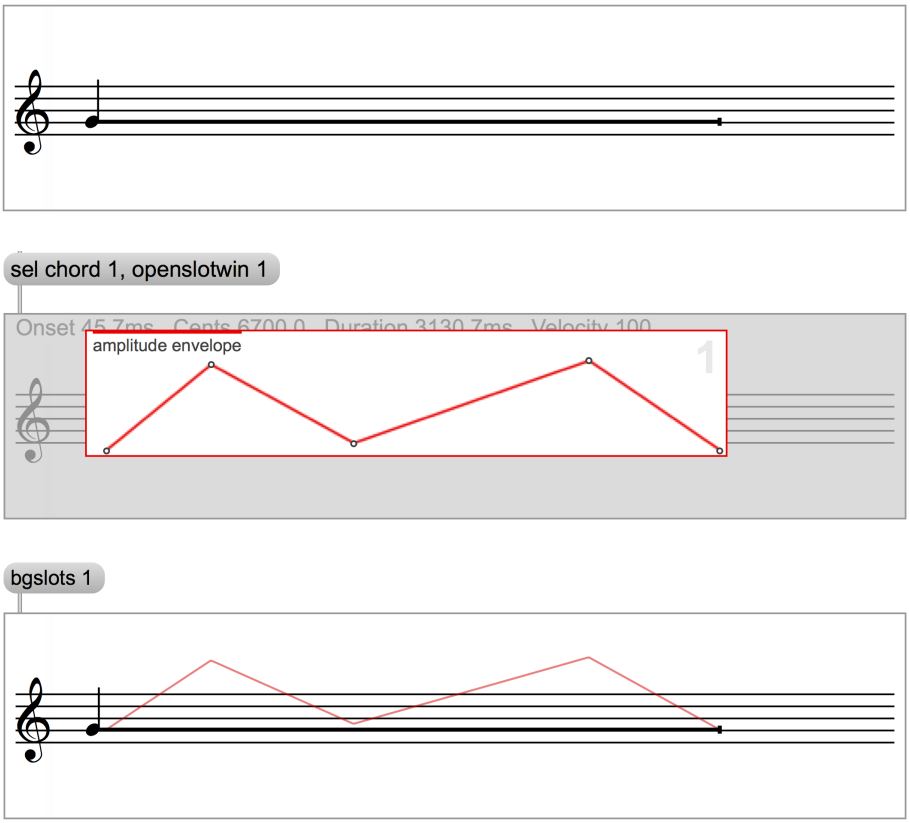
- *int* — Zum Darstellen und Ausgeben von einzelnen ganzen Zahlen.
- *float* — Zum Darstellen und Ausgeben von einzelnen Fließkommazahlen.
- *intlist* — Zum Darstellen und Ausgeben von *Listen* von ganzen Zahlen.
- *floatlist* — Zum Darstellen und Ausgeben von *Listen* von Fließkommazahlen.
- *function* — Zum Darstellen und Ausgeben von linearen Funktionskurven.

Durch die Angabe eines *slope*-Werts kann auch jedes lineare Segment zu einer Kurve werden.

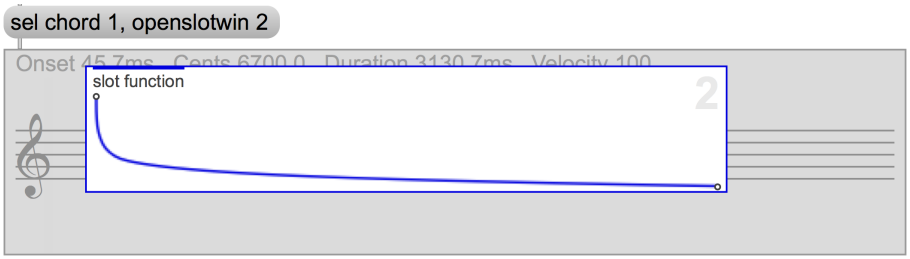
- *spat* — Zum Darstellen und Ausgeben von *Panorama*-Werten.
- *text* — Beliebiger Text zur einzelnen Note.
- *llll* — Ganze *bach*-Partituren – auch kleinere *Listen* von *Listen* – können in einer Note verpackt werden.
- *filelist* — Informationen über *Files* auf der Computer-Festplatte.
- *color* — Farbdarstellung im *RGBA* – Modus.
- *3dfunction* — Lineare Kurve mit x y z – Koordinaten für jeden Punkt. Auch hier sind zusätzliche *slope*-Werte durch einen vierten Parameter pro Funktions-Punkt möglich.
- *filter* — Einstellbare Filtertypen: *lowpass*, *highpass*, *bandpass*, *bandstop*, *peaknotch*, *lowshelf*, *highshelf* und *allpass*. Frequenzangabe in Hertz und *Gain* in Dezibel. Q ist der *resonance*-faktor oder *slope*-faktor – abhängig vom Filtertyp.
- *dynfilter* — Beinhaltet Informationen über die zeitliche Abfolge von *interpolated biquad filters*. Diese Filter-Parameter sind in erster Linie zur internen MAX (vormals MSP)-Weiterverwendung nützlich.
- *togglematrix* — Zahlen in Reihen und Spalten organisiert. Jeder Eintrag als *Toggle*-Information – Ein oder Aus.
- *intmatrix* — Jeder Matrizen-Eintrag als ganze Zahl.
- *floatmatrix* — Jeder Matrizen-Eintrag als Fließkommazahl.

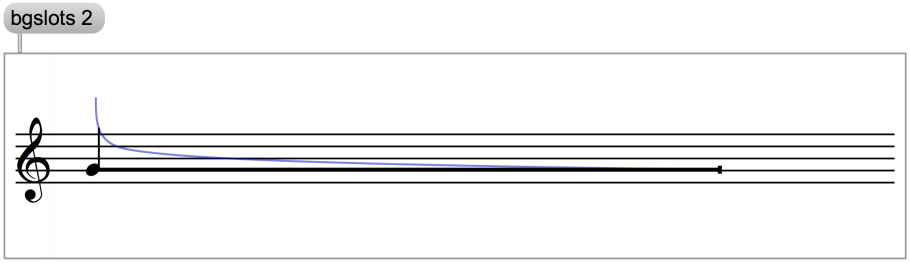
Willkommen in der Zukunft.

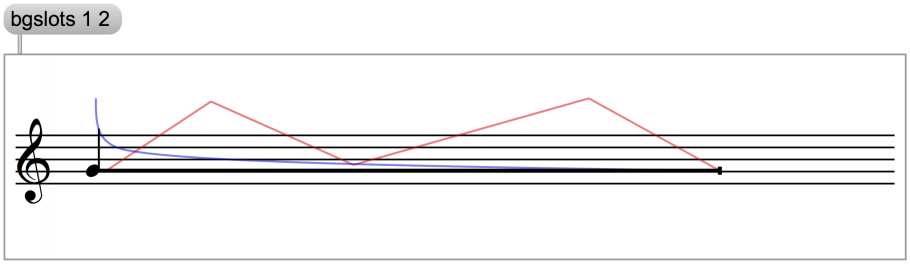
Man darf gespannt sein, wie diese neuen flexiblen Möglichkeiten der Notation von Musik-Programmierern aufgenommen werden. Und ob sich ihre Ergebnisse wiederum auf die traditionelle Notation auswirken werden.



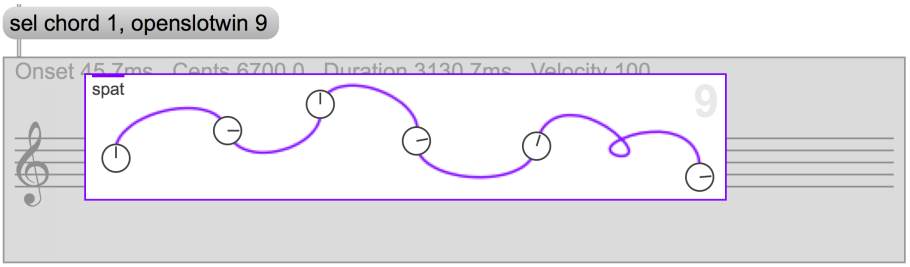

Durch Öffnen eines *slot*-Fensters (Anwählen einer Note + definiertes Tastaturkürzel) können je nach *type* die Funktionsparameter eingegeben werden. Natürlich ist es wieder möglich, sämtliche Parameter auch mit *Messages* festzulegen.

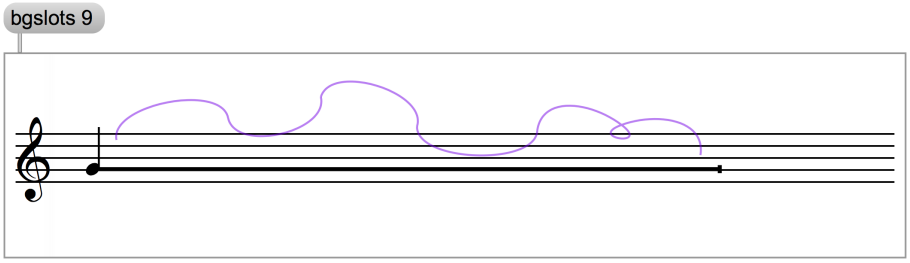


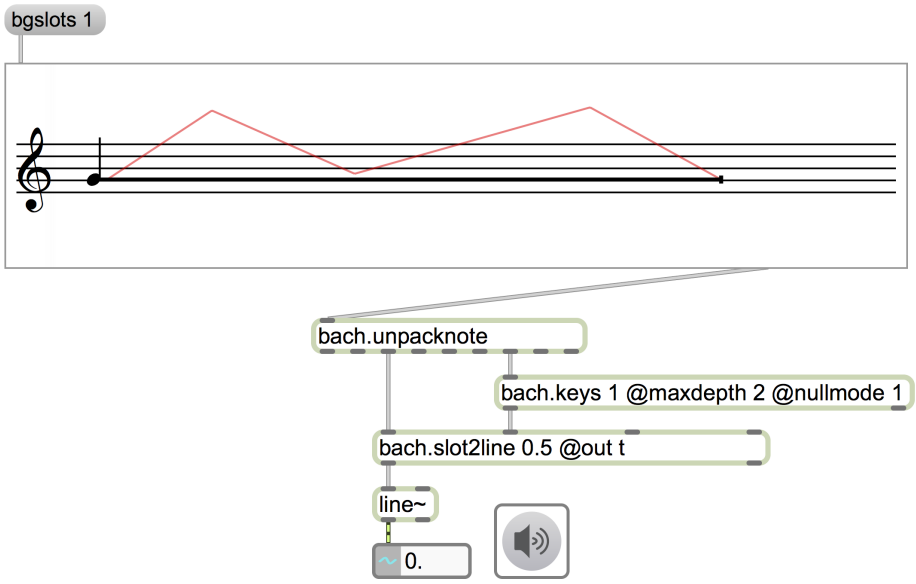



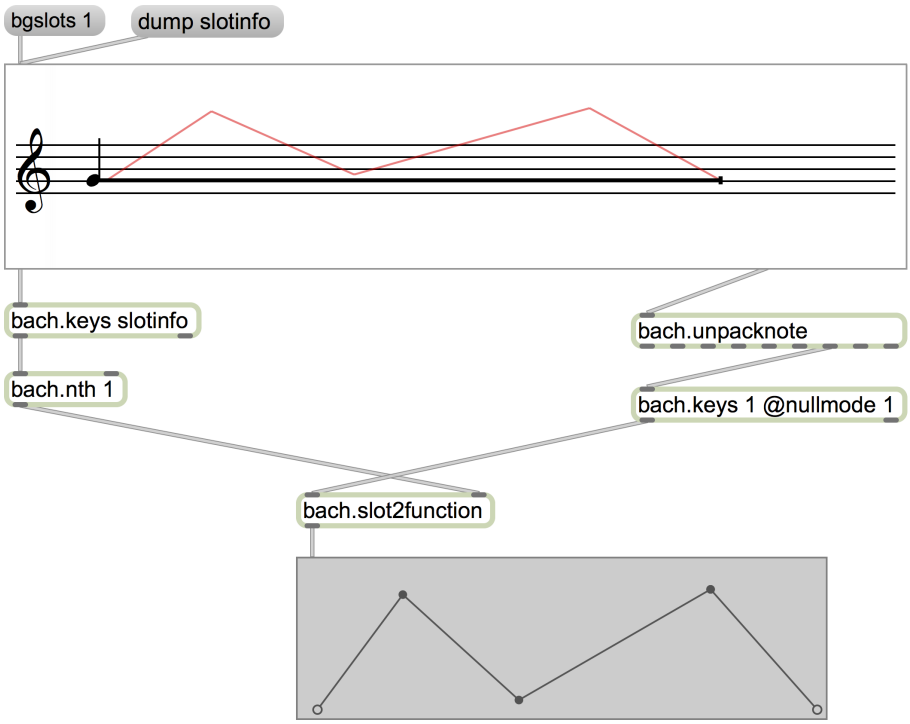

Durch die *Message bgplots*+Zahl(en) werden die entsprechenden *slots* auch im Notenbild dargestellt.



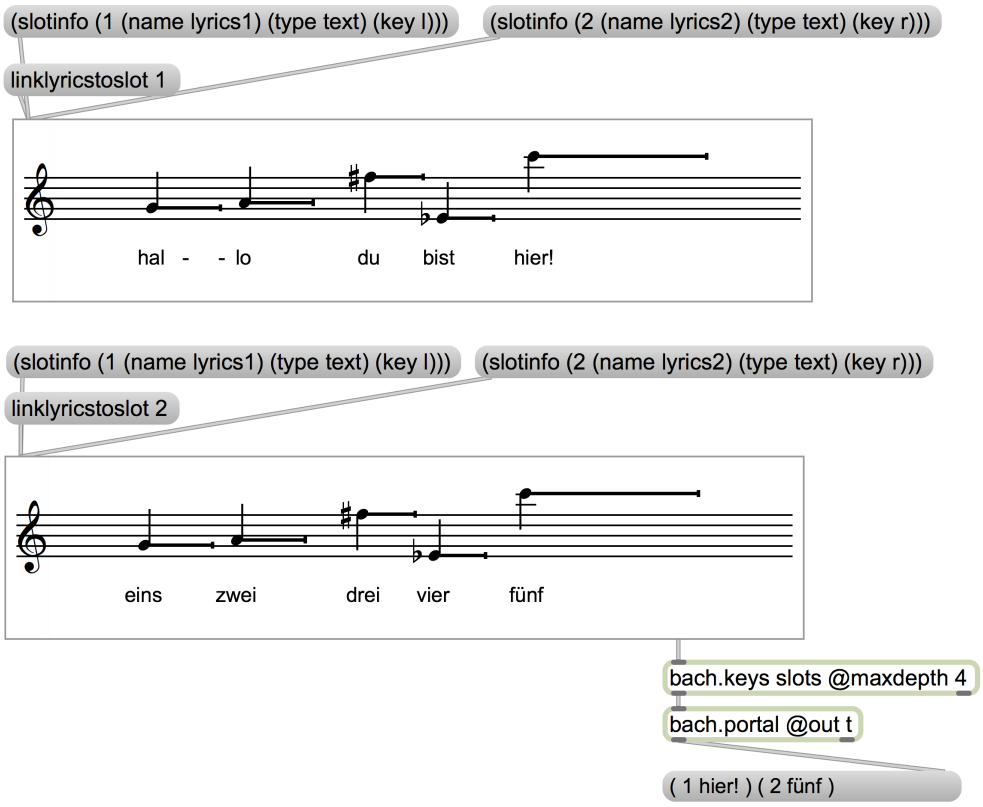
Die Daten der *slots* werden mit geeigneten OBJECTS „abgefangen“ und der weiteren Prozessierung zur Verfügung gestellt.



Selbst der Umgang mit Text beziehungsweise *Lyrics* wird in *bach* mit *slots* gelöst:

1. Note selectieren
2. Cmd+Shift+L zum Beginnen der Texteingabe
3. Mit Tab zur nächsten Note

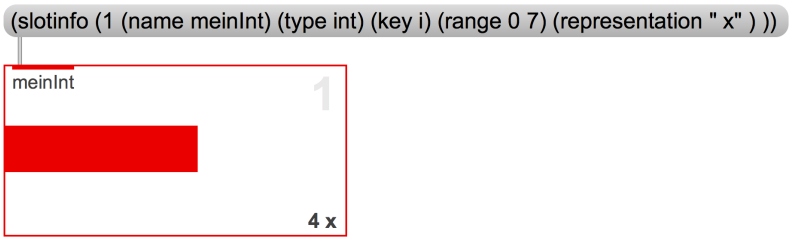



Praktisch können auch mehrere *slots* für Text bereitgestellt werden.
Mit der *Message linklyricstoslot*+Zahl wird der entsprechende *slot* mit dem dazugehörigen Text im Notenfenster von *bach.score* oder *bach.roll* angezeigt und bei Bedarf auch ausgegeben.

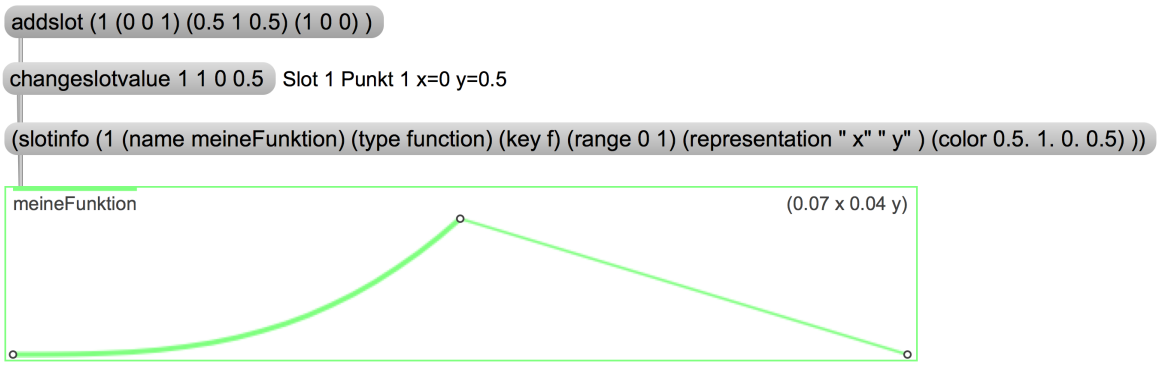
Die *slots* werden neben der *Message* für die *slot*-Nummer mit

- frei zu wählenden Namen (*name*),
- der Kategorie (*type* von *int* bis *llll*),
- dem frei zu wählenden Tastenkürzel zum Aufrufen des *slot*-Fensters (*key*),
- dem Wertebereich (*range* – je nach *type* unterschiedlich),
- der Repräsentanz der Werte (*representation* – zum Beispiel Maßeinheiten oder Ähnliches
- oder mit der Farbe des *slot*-Fensters

definiert.

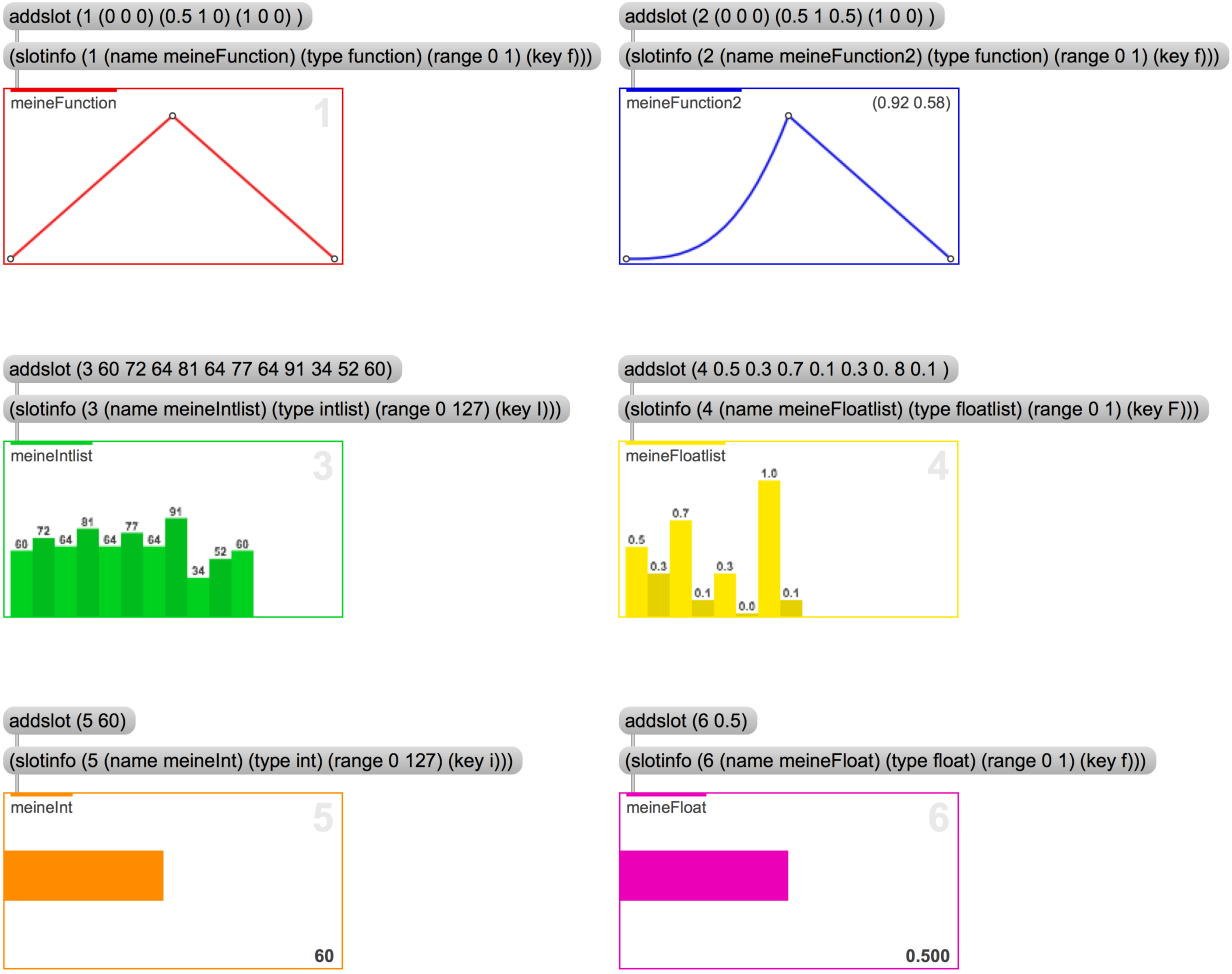



Neben der direkten Eingabe im *slot*-Fenster wird mit den *Messages addslot* oder *changeslotvalue* die Werteingabe und -veränderung unterstützt.



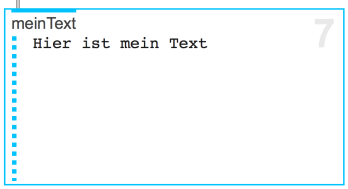
## Abschließend

noch eine Übersicht mit *Minimal*-Beispielen der verschiedenen *slot*-Variationen in eigenen *slot*-OBJECTS.
Die ersten zehn *slots* sind in ihrer Kategorie vordefiniert – der Musik-Programmierer darf aber auch frei über jeden *slot* verfügen.



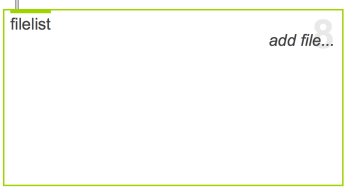
addslot (7 " Hier ist mein Text" )

(slotinfo (7 (name meinText) (type text) (key t)))



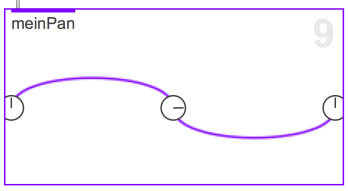

addslot (8 )

(slotinfo (7 (name meinText) (type text) (key t)))



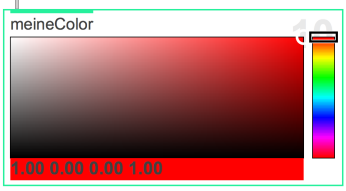

addslot (9 (0 5 0 0 ) (0.5 5 90 0) (1 5 0 0) )

(slotinfo (9 (name meinPan) (type spat) (key p)))




addslot (10 1. 0. 0. 1. )

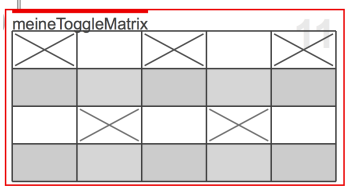
(slotinfo (10 (name meineColor) (type color) (key c)))




addslot (11 ((1 0 1 0 1 ) () (0 1 0 1 0 ) () ) )

(slotinfo (11 (representation (4 5 ))))

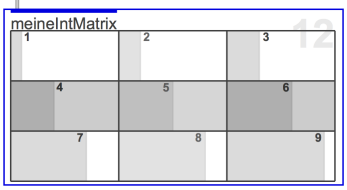
(slotinfo (11 (name meineToggleMatrix) (type togglematrix) (key m)))




addslot (12 ((1 2 3 ) (4 5 6) (7 8 9 )) )

(slotinfo (12 (representation (3 ))))

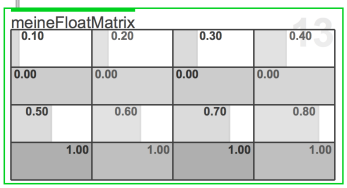
(slotinfo (12 (name meineIntMatrix) (type intmatrix) (range 0 10) (key y)))




addslot (13 ((0.1 0.2 0.3 0.4 ) () (0.5 0.6 0.7 0.8 ) (1. 1. 1. 1. )) )

(slotinfo (13 (representation (4 ))))

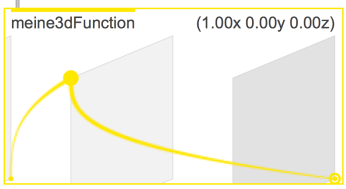
(slotinfo (13 (name meineFloatMatrix) (type floatmatrix) (range 0 1) (key v)))



addslot (14 (0 0 0 0 ) (0.5 1 1 -0.5) (1 0 0 -0.5))

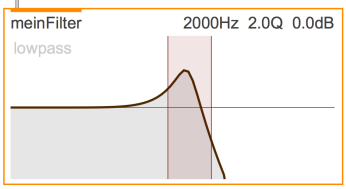
(slotinfo (14 (representation x y z )))

(slotinfo (14 (name meine3dFunction) (type 3dfunction) (range 0 1) (key z)))

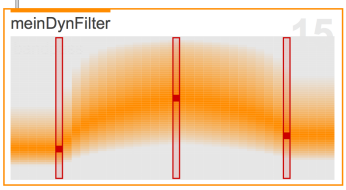



addslot (15 lowpass 2000 0. 2.)

(slotinfo (15 (name meinFilter) (type filter) (range -12 12) (key u)))

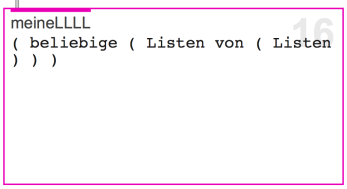


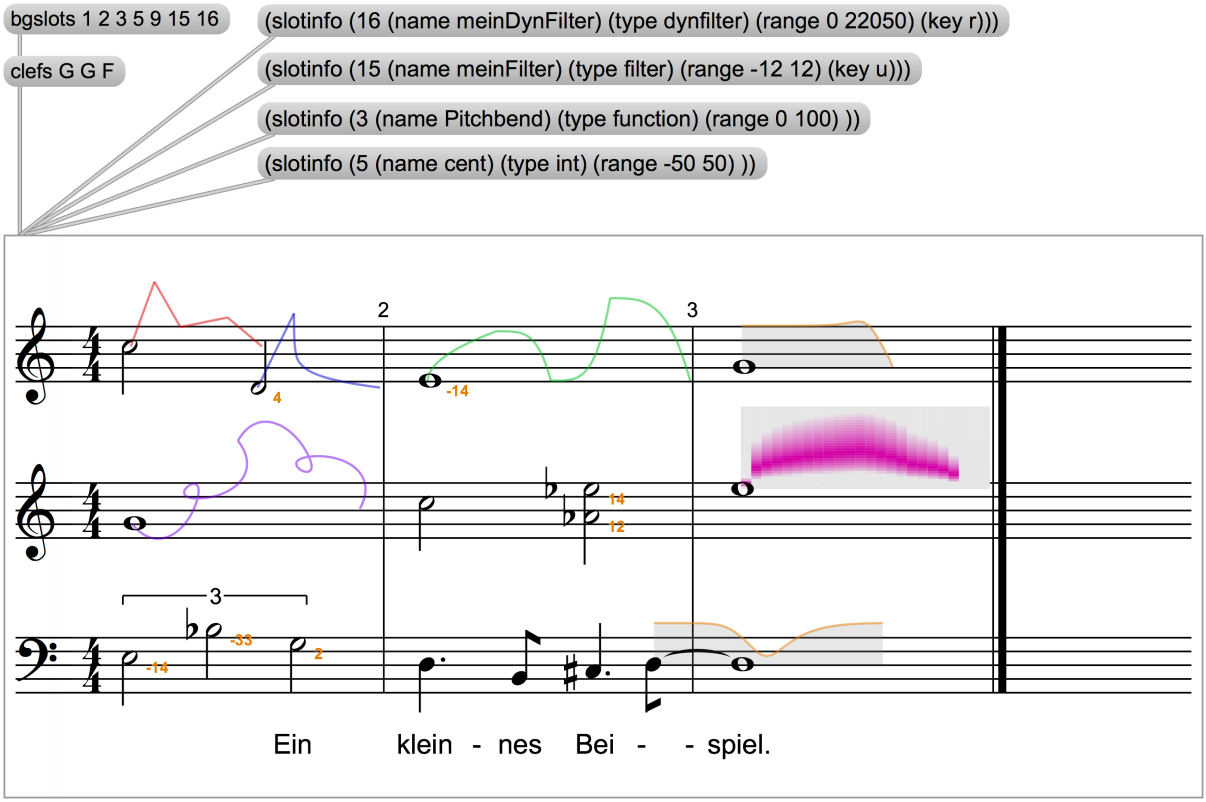

(slotinfo (16 (name meinDynFilter) (type dynfilter) (range 0 22050) (key r)))




addslot ( 17 ((beliebige (Listen von (Listen)))))

(slotinfo (17 (name meineLLLL) (type llll) (key l)))




Jeder einzelnen Note stehen sämtliche *slots* zur Verfügung.

Ein zusammenfassendes Beispiel von *slot*-Darstellungsvarianten in *bach.score*: