

Gabriele Petrillo

Fondamenti di Max

Questa prima lezione è dedicata allo studio dei fondamentali della programmazione in Max Msp.

01 Patchcords

Max è un linguaggio di programmazione basato sul collegamento di oggetti grafici tramite cavi chiamati *patch chords*. Gli *object box* sono piccoli programmi che eseguono delle azioni di calcolo, tuttavia in Max sono presenti altre due tipologie di scatole chiamate *message box* e *comment box*: i primi sono messaggi alfa-numeriche che possono essere inviati o ricevuti da un object box, i secondi sono commenti che possiamo scrivere all'interno della patch. A seconda del tipo di dati che gli object box spediscono i cavi di collegamento saranno di colore diverso: **grigi** dati numerici o stringhe, **gialli** per flussi di dati audio e **verdi** per il video.

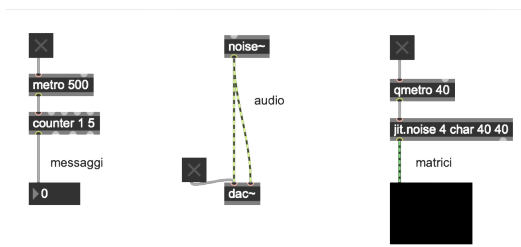


fig.1

02 Introduzione a Max

Il **bang** è un messaggio che ha il compito di dire agli oggetti di fare quello che devono fare, quindi inviare dei messaggi dai loro outlet. Può essere collegato a più oggetti o messaggi per ottenere un output multiplo. L'oggetto **print** serve a visualizzare l'output di un messaggio o di un oggetto nella console di max, può essere chiamato con un nome specifico, ed ha una funzione di controllo. Nella prima patch un **bang** è collegato a tre messaggi, collegati a loro volta all'oggetto **print A**. Se facciamo click sul bang possiamo vedere stampati nella console di

Max i tre messaggi. Gli oggetti **bangbang** e **trigger** servono ad indirizzare più messaggi di bang nel primo caso, messaggi vari nel secondo, in un certo ordine. L'argomento dell'oggetto **bangbang** è il numero di output desiderato, mentre quello di **trigger** è il tipo di messaggi che deve inviare in un determinato ordine. Una differenza importante è che l'oggetto **trigger** può ricevere un qualsiasi tipo di messaggio ed inviarlo da qualsiasi output implementato.

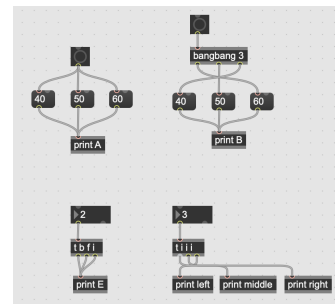


fig.2

03 Number Box

Il **number box** è un oggetto che invia dei messaggi contenenti dei numeri. Esistono due tipi di Number box: **float** (numeri con la virgola) **int** (numeri interi). I numeri possono essere controllati tramite scroll, qualsiasi oggetto grafico come dial o fader, o scrivendoci dentro il numero. Ogni volta che cambiamo inseriamo un numero l'oggetto provoca un output del numero, tuttavia se vogliamo solo caricare il numero senza provocare nessun output bisogna inviare un messaggio con in numero preceduto dalla keyword *set*. Se colleghiamo un Number box ad un messaggio, possiamo avere un messaggio con un numero variabile inserendo nel messaggio la keyword "\$1", per i numeri interi o "\$1." per i numeri con la virgola. La conversione di un numero di un tipo in un altro si

chiama casting. Se convertiamo un numero int in un float, la conversione è senza perdite di informazione, viceversa perderemo tutti i numeri dopo la virgola.

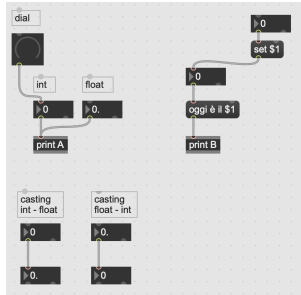


fig.3

04 Toggle e Metro

L'oggetto **toggle** di default invia uno 0 o un 1 e spesso è utilizzato come un interruttore per gli molti oggetti max. Tuttavia se inviamo un numero tramite messaggio a **toggle** questo lo lascerà passare facendo un casting in integer. L'oggetto **metro** è un oggetto molto utile in max ed è utilizzato per sincronizzare il funzionamento di più oggetti e regolare la velocità di generazione degli eventi. Sostanzialmente è un metronomo che si avvia con un messaggio di 1 e si spegne con 0. L'output dell'oggetto **metro** sono una serie di bang ad una velocità regolata per default in millisecondi e che può essere specificata o come argomento, o con un messaggio nell'inlet destro. Se colleghiamo più oggetti metro ad uno stesso messaggio di start possiamo azionare processi a velocità diverse parallelamente.



fig.4

05 Ordine degli eventi e debug

Anche se può sembrare che max faccia più cose contemporaneamente, tutti i calcoli vengono fatti uno alla volta e in un certo ordine. È molto importante mettere gli oggetti nel giusto posto nella patch per far capire a max in che ordine fare i calcoli. L'ordine di precedenza di max è destra - sinistra, basso - alto. Se ci sono

dubbi è possibile verificare l'ordine dei messaggi con la funzione debug (nel menù in alto) applicando i watchpoint sui cavi. (per fare questo dobbiamo andare nel menù debug ed abilitare il debug, poi selezionare un cavo ed inserire il watchpoint desiderato) Per ovviare al problema grafico del posizionamento degli oggetti possiamo usare l'oggetto "trigger" che dato un certo input (qualsiasi messaggio alphanumerico) può gestire l'ordine di output di determinati eventi.

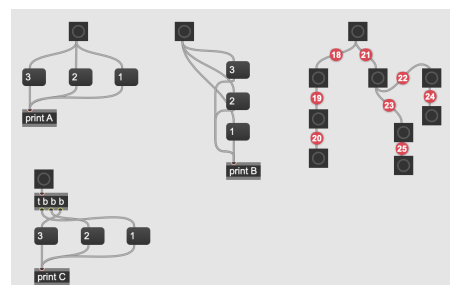


fig.5

05 Liste

Per lista si intende un messaggio contenente una serie di dati alfanumerici. Le liste sono utili per inviare variabili con Keyword agli oggetti o far camminare più informazioni contemporaneamente per poi essere divise. Gli oggetti che ci permettono di creare o spaccettare liste sono "pack" "pak" e "unpack". I primi due servono a creare liste, come argomento vanno inseriti i tipi di dato che devono ricevere dagli inlet rispettivi, la differenza tra i due oggetti è che il primo ha solo il primo inlet caldo, mentre il secondo gli ha tutti. L'oggetto "unpack" serve a spaccettare una lista in più messaggi singoli e come argomento richiede i tipi di messaggi che deve spaccettare dall'outlet rispettivo. Per creare dei messaggi contenenti delle liste possiamo utilizzare gli oggetti "append" e "prepend" che come argomento vogliono il messaggio da inserire dopo o prima del messaggio arrivato dall'inlet. Se nella messaggio sono contenute virgole, Max interpreterà quel messaggio come due stringhe.

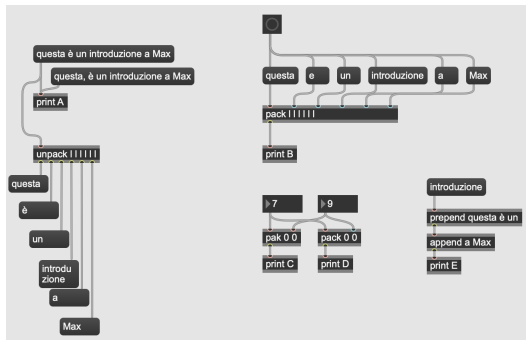


fig. 6

BIBLIOGRAFIA

- CIPRIANI, ALESSANDRO AND GIRI MAURIZIO, *Musica Elettronica e Sound Design*, Contemporinet 2012