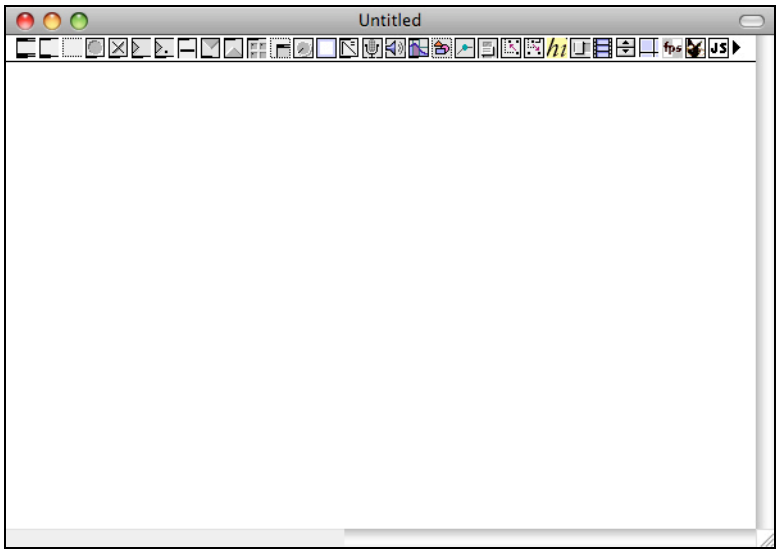
F1. FÜGGELÉK: INTERAKTÍV RENDSZEREK ELŐKÉSZÍTÉSE MAXMSP/JITTER KÖRNYEZETBEN

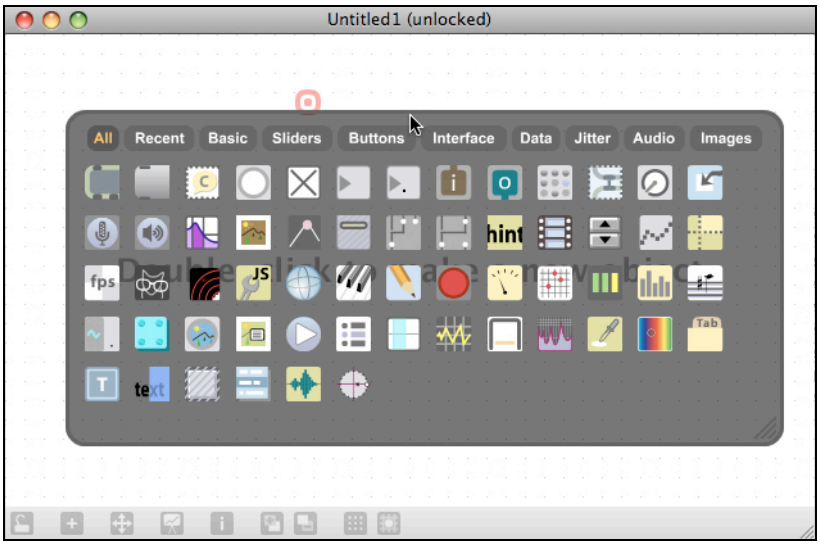
A MaxMSP mint metaalkalmazás

A MaxMSP nem nevezhető alkalmazásnak abban az értelemben, amennyiben az alkalmazás fogalma egy, programozók által kijelölt feladat felhasználó által elindított végrehajtását jelenti. Jelen esetben csupán egy üres keretrendszert kapunk, amelyben az előre ki nem jelölt korlátok között kitalálhatjuk és megvalósíthatjuk a feladatokat. Minden egyéb képessége mellett a szóban forgó alkalmazás elsősorban multimédia célokra hegyezhető ki. Ezek között a leghagyományosabb az élő MIDI-kezelés és hangszintézis, videójel-feldolgozás. A Max története is ennek megfelelően alakult. 1990-ben kezdte fejlesztését a párizsi IRCAM (Institut de Recherche et Coordination Acoustique/Musique). Nevét *Max Matthews* amerikai mérnökről kapta, aki sok egyéb, a számítógépes zenét megtermékenyítő kezdeményezése mellett az első zenei program kreátora. Ezek az ún. musicN nyelvek, ahol az N különböző verziószámokat jelöl, és általános jellegzetességük, hogy szöveges, nem valós idejű hangszintézist és algoritmikus komponálást tesznek lehetővé, ugyanakkor mindezt a technikai korlátok közepette is igen hatékonyan. Nem véletlen ezért, hogy a CSound, ezen programnyelvek egyik legkifinomultabb változata, a mai napig a számítógépzenei diskurzus alapvető része (ROADS 1987, CMT stb). A Max kezdeti verziói is mutatták, hogy lényegi váltás következik be: a felhasználó ugyanis *grafikus* felületen, a programozás és elektronikai eszköztár metaforáit digitális úton kezelheti, így a legkönnyebben végezhetett egyszerű aritmetikai műveleteket – a kezdeti állapotban csupán MIDI-jeleken, ezt követően, az MSP (Max Signal Processing) objektumok 1997-es megjelenésével már valós idejű hangokon, majd a Jitter kiterjesztés megjelenésével már videójeleken is. Jelenleg e három modul már egyazon egységet jelent, és a képességeit számos önkéntes fejlesztő az igényeknek megfelelően bővíti.



f/1a. ábra: a MaxMSP/Jitter felületének alapja, az üres patch

A felső menüsoron látható főbb elemek rendre: objektum , üzenet , megjegyzés , nyomógomb (`button`), kapcsoló (`toggle`), egész- és törtszámos számdoboz / (`number` és `flonum`)

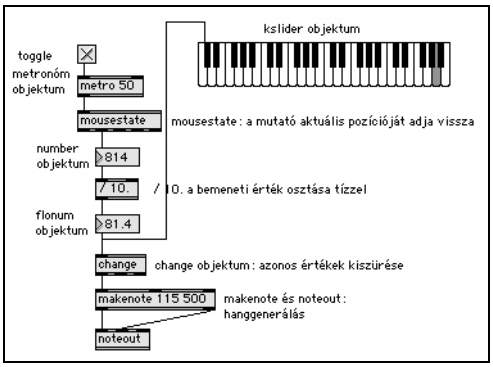


f/1b. ábra: a MaxMSP/Jitter 5-ös verziójának felülete
Az f/1a. ábrán látható menüsor itt duplakattintással jeleníthető meg. Újdonság az objektumok kategorizáltsága, valamint a patcher alsó során található prezentációs és navigációs funkciók.

A Max működése az analóg szintetizátorokban működő fizikai szabályok mellett az objektumorientált programnyelvekre hasonlít, ami azt jelenti, hogy a feladatokat végrehajtó objektumok parancsok (üzenetek) által befolyásolhatók, s több, komplex feladatot ellátó objektum önálló objektumként is (ennek neve absztrakció) hivatkozható. A Max emellett *élő*, vagyis a feladat végrehajása közben is módosítható a futtatás alapját képező *patch*. Az f/1a és f/1b ábrák mutatják az üres patch állapotát. A menüsorból előhívható objektumok összekapcsolásuk után egymásra hatnak

Max és a kontrolljelek

A kontrolljelek olyan üzenetek, amelyek nem magát a megszólaló hangot vagy megjelenő képet képezik le, hanem az ahhoz kötődő események megtörténtét és paramétereit jelzik. A legegyszerűbb kontrolljel a bang (`button` [ ]), a szám (egész- vagy törtszám; `number` [43], `flonum` [2.27]) és az üzenet (`message` [üzenetdoboz]). A Max keretein belül minden típusú feladat leképezhető ezen kontrolljelek szintjén, hiszen az összes grafikus vezérlőelem (pl. potméterek, klaviatúrák, kapcsolók) számokat küldenek és fogadnak, az időzítést kezelő metronóm (`metro`) a megadott időközönként bang-üzeneteket bocsát ki magából stb. A kontrolljelek ilyetén leképezése az üzenetek hatékony manipulálását és automatizálását is lehetővé teszi, ugyanakkor ezek kezelése igényli a legkevesebb számítási kapacitást a számítógép részéről. Az alábbiakban a Max és a külvilág között közvetítő kontrolljelek szabványait tekintjük át.



f/2. ábra: szintetizátorvezérlés egérrel

A patch működésben mutatja a főbb objektumegységeket. A kapcsoló a metronómot vezérli, a metronómból érkező „bang" a mousestate objektumot készteti a mutató pozíciójának kiolvasására. Ez az érték a `mousestate` *2. kimenetén jelenik meg, melyet a* `/ 10.` *kisebb tartományba redukál, majd a* `makenote` *és* `noteout` *szólaltat meg. A művelet akkor is megszólaltatja az egér mozgását, amikor a számítógépen a Max alkalmazás a háttérben fut, így lehetséges a felhasználói felület hangzó kiterjesztése.*

1. MIDI

A MIDI (Musical Instrument Digital Interface, hangszerek digitális kapcsolata) szabvány 1982-ben született, amikor az elterjedt szintetizátorok és elektronikus hangszerek (szekvenszerek, mintavevők stb.) felvetették a szinkronizálhatóság és vezérelhetőség igényét. Az addig létező CV (control voltage, vezérlőfeszültség) kereteit kinőve nemcsak arra volt szükség, hogy a hang leütését és hangerejét, hanem hangmagasságát és egyéb a hangra jellemző tulajdonságokat is vezérelni lehessen. A MIDI adatfolyamai nem a hangokat, hanem a hangok megszólalásának kontrollparamétereit közvetítik. A MIDI elterjedése által lehetővé vált, hogy a hangot kibocsátó hangszer és az azt megszólaltató felület különváljék, ez utóbbi akár több, fizikai távolságban is elhelyezkedő hangszert is képes legyen vezérelni. Az A/2 melléklet tartalmazza a Max MIDI objektumait, és e listából is előtűnik, hogy lehetővé teszi az adatfolyam alacsony (`midiin` és `midiout`) és magas (pl. `borax`) szintű kezelését.

A MIDI szabvány 16 csatorna (vagyis maximum 16, egymás után sorba kötött hangszer) különálló vezérlését teszi lehetővé. Ez a szám látszatra igen magas, de a szabvány számos egyéb hátránya (lassú adatátvitel, a hálózaton keresztül megvalósítható kommunikáció hiánya) miatt újabb szabványok is megjelentek mellette.

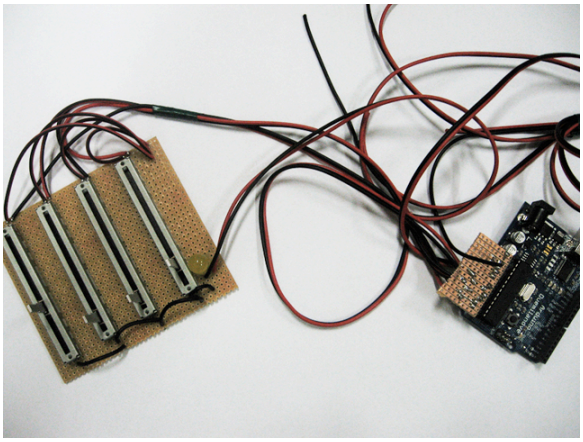
2. OSC

Az OpenSoundControl kitágította a MIDI által létrehozott fizikai kapcsolatot, az ethernet hálózati kommunikáció által már bejáratott TCP/IP alapú protokoll által mindenfajta, hálózatba kapcsolható eszköz számára lehetőséget biztosít a kommunikáció számára. Az OSC hálózati címek és adatstruktúrát formájában továbbítja a kontrollüzeneteket. Mivel jelenleg kevés hangszer használja ki képességeit, ezért számítógép és számítógép közötti kapcsolatteremtése a Max által biztosított `udpsend` és `udpreceive` objektumok használhatók, melyek – igény szerint – az OSC protokollt is közvetíteni képesek.

3. Soros kommunikáció

A soros kommunikáció által a Max minden, a számítógéphez csatlakoztatott perifériának képes üzeneteket küldeni és visszajelzéseket fogadni. Két kurrens példa ezek között a DMX fényvezérlő szabvány, valamint az Arduino interfész kezelése. A

DMX elsősorban fényvezérlő eszközök számára kifejlesztett, a MIDI-nél egyszerűbben kezelhetőbb, de igen hatékony és nagy kapacitású szabvány. Az eredeti specifikáció szerint 512 csatornát kezel, csatornánként 0-255 közötti értékeket továbbít. A DMX-jelek kezelése a `dmxusbpro` (fejleszti Olaf Matthes[XXX]) objektummal lehetséges, külső konverter-egységen keresztül. A nyílt fejlesztésű *Arduino* interfész elérése kizárólag felhasználói szintű beavatkozást tesz lehetővé. Az ARDUINO2MAX[XXXI] és FIRMATA/MAXDUINO[XXXII] absztrakciók az interfész által biztosított ki- és bemenetek adataihoz engednek hozzáférni. A hozzáférés kétirányú: nemcsak az Arduino-hoz csatolt potméterek, szenzorok állapota olvasható ki, hanem az interfészre kötött diódák és motorok is vezérelhetők. Az Arduino újabb, tesztelés alatt álló verziót vezetéknélküli (Bluetooth) kommunikációt is lehetővé tesznek.



f/3. ábra: Arduino interfész és hozzá kapcsolódó tolópotméterek
Az interfész elemei mindennapi elektronikai eszközökből összeállíthatók.

4. GUI

Amennyiben nem áll rendelkezésre külső interfész, hangszer, a fenti protokollok valamelyikét kezelő eszköz, úgy természetesen a számítógép beépített eszközei is felhasználhatók egy- vagy kétirányú kommunikációra. A `mousestate` objektum az egérmozgást és az egér gombjait, görgőit alakítja vezérlőjelekké (lásd f/2. ábra), a `key` objektum a billentyűzet gombjaihoz rendelhet feladatokat. Az *aka* objektumok (fejleszti Masayuki Akamatsu[XXXIII]) a számítógép belső szenzoraiból (hő-, fény-, mozgásérzékelő stb.) nyert információkat továbbítják a Max felé.

Hang be- és kimenet

A MaxMSP környezetben a valós idejű hangfeldolgozásra vonatkozó objektumok konvenció szerint „~“ (tilde) jellel végződnek. A megkülönböztetés a feldolgozási folyamatok hatékonyabb kihasználása végett szükséges, és hatékonyságát növeli a kontrolljelek felé irányuló kétirányú átjárási lehetőség, mint pl. a hangjelek kontrolljellé alakítása (`meter~` v. , `snapshot~` vagy az `analyzer~`[XXXIV]) és fordítottja (a `sig~`). Hangok feldolgozása események vezérlése céljából szintén erre az átjárásra épül, vegyünk csak egy olyan egyszerű példát, mint a hangerőre reagáló felvillanás: a mikrofonon (`adc~` v. ) keresztül érkező hangjel egy kivezérlésjelzőn (`meter~`) keresztül hangerőértéket kap, amelyre egy feltételes függvényt csatolva megszólaló hangot vezérelhetünk. A megszólaló hang lehet egy midi hangszer (`makenote` és `noteout` kombináció, lásd f/2 ábra), vagy a MaxMSP által generált hang, mint pl. egy színuszjel (`cycle~`) vagy egy hangminta (`buffer~` és `groove~`

kombináció), amelyek a digitális-analóg konverteren (`dac~` v. ) keresztül szólalnak meg.

Videójel-kezelés

Míg videóállományok és képek megjelenítését a Max alapszinten is támogatta, addig a Jitter objektumok a videójelek kifinomult használatát teszik lehetővé. A Jitter számára a vizuális információ többdimenziós mátrixokban kezelhető. A mátrixok nemcsak a színcsatornákat, hanem összesen akár 32 réteget [plane] is jelenthetnek. Az egyes rétegek általános jellemzői: felbontás (horizontális pixel x vertikális pixel) és az adat típusa (egészszám, lebegőpontos szám, karakter). A Jitter objektumainek megnevezése „jit." kifejezéssel kezdődik, és az adatok az MSP-vel ellentétben nem folyamatosan, hanem bang-jelzések hatására kerülnek továbbításra. Így pl. egy 25 fps sebességgel meghatározott videójel-rendszert a `metro 40` (1000 ms / 25 fps = 40 ms) objektum tarthat megfelelő működésben. A folyamatosság ellenében szól a tény, hogy a Jitter igen nagyméretű adattömegek mozgatását végzi, így a felhasználónak folyamatosan érdemes felülbírálnia a képfeldolgozás valósidejű jellege és a rendszert lelassító hatása között fennálló kapcsolatot.

A videójel beolvasását a `jit.qt.grab` (Windows és Mac esetén) v. `jit.dx.grab` (Windows esetén) objektumok végzik. Amennyiben nem áll rendelkezésre kamera, úgy videóállományokat is beolvashatunk a `jit.qt.movie` segítségével. Mindegyik lehetőség esetén érdemes megadnunk a feldolgozandó kép méretét (alapértelmezett esetben 320 x 240), mivel CV feladatokhoz alacsonyabb felbontás is elegendő, videóprezentációhoz pedig nagyobb érték használatos. A videójel monitorozása a patcher ablakában `jit.pwindow`, attól függetlenül a `jit.window` objektum által lehetséges. Ha a monitorablak fekete, nem érkezik jel, s így meg kell nyitni a kamerát (open parancs) vagy be kell tölteni egy videóállományt (read parancs). Ezt követően a videójel a 4.2.2. fejezetben taglaltak szerint hasznosítható.

F1/a. melléklet: alapvető Max-objektumok

Az alább következő táblázatok a MaxMSP/Jitter alkalmazáshoz kapcsolódó referenciagyűjtemény[XXXV] kivonatolt és funkció szerint rendezett változatai.

Objektum	Kategória	Leírás
+, -, *, /, pow, sqrt	Aritmetika	alapvető aritmetikai műveletek: összeadás, kivonás, szorzás, osztás, hatványozás, gyökvonás
<, <=, ==, !=, >=, >, !=	Aritmetika	relációs operátorok: kisebb mint, kisebb v. egyenlő mint, egyenlő, nem egyenlő, nagyobb vagy egyenlő mint, nagyobb mint
abs	Aritmetika	abszolútérték
button, ubutton	Kontroll	gomb, bang-et ad ki
change	Kontroll	kiszűri az ismétlődéseket
clip, scale	Kontroll	határértékek közé szorít be számokat
coll	Adattárolás	adatokat tárol és szerkeszt
counter	Aritmetika	számláló
cycle	Kontroll	számokat cirkukáltat a kimeneteken
date	Rendszer	a rendszerdátumot és -időt adja vissza

delay (del)	Kontroll	bang üzenetek késleltetése
float (f), int (i)	Adattárolás	számok tárolása
function	Adattárolás	grafikus töréspontszerkesztő
gate, route, router	Kontroll	üzenetek irányítása
hslider, vslider, multislider, slider, dial, pictslider, rslider, uslider	Interfész	csúszkák, potméterek
if ... then ... else	Aritmetika	ha .... akkor .... különben feltételes ciklus
iter	Kontroll	listák átalakítása sorozattá
key, keyup, modifiers, numkey, mousestate, mousefilter	Interfész	billentyűzet és egéradatok
kslider, nslider	Interfész	klaviatúra
line, linedrive	Aritmetika	interpoláció (értékek kiegyenlítése)
maximum, minimum	Aritmetika	maximum / minimum feltétel
message	Kontroll	üzenet
metro, tempo	Kontroll	metronóm
number box	Kontroll	számüzenet (nem objektum!)
pack, pak, unpack, unpak	Kontroll	bemeneteket egy listára fűz / szétbont
panel	Interfész	háttérfelület rendező
patcher (p)	Interfész	al-patcher
peak	Aritmetika	megtalálja egy sorozat legnagyobb értékét
pipe	Kontroll	számok és listák késleltetése
prepend, append, sprintf	Kontroll	üzenetek rendezése
preset	Adattárolás	patcher beállításainak elmentése, visszahívása
print	Interfész	üzenetek kiírása a rendszerablakba
random, urn	Aritmetika	véletlenszám-generálás
select (sel)	Aritmetika	megadott üzenetek és számok kiválasztása bejövő üzenetek közül
send (s), receive (r)	Kontroll	üzenetek továbbítása kötések nélkül
split	Kontroll	üzenetek szétbontása tartományokra
table	Interfész	adatok tárolása és grafikus szerkesztése
toggle	Interfész	kapcsoló (értéke 1 v. 0.)
trigger (t)	Kontroll	beérkező üzenetek szétosztása típusok szerint
ubumenu, umnu	Interfész	legördülő menük
zl	Aritmetika	számsorozatok elemzése

F1/b. melléklet: alapvető MIDI objektumok

Objektum	Kategória	Leírás
bendin, bendout, xbendin, xbendout	MIDI	hajlítás kezelése
borax	MIDI	midi hangjegyek elemzése
ctlin, ctlout	MIDI	kontroller üzenet be/ki
detonate	Adattárolás	grafikus kotta- és eseményszerkesztő
flush, midiflush	MIDI	note-off generálás a tartott hangokon
ftom, mtof	Konverzó	frekvencia-midi hangjegy konverter
makenote	MIDI	automatikus note-off
midiformat, midiparse	MIDI	midi üzeneteket formáz / elemez
midiin, midiout	MIDI	alacsonyszintű midi kezelés
mtr	Adattárolás	többsávos adattárolás
notein, noteout	MIDI	hangjegyek küldése, fogadása
pgmin, pgmout	MIDI	programváltás be / ki
poly, polyin, polyout	MIDI	polifón hangok szétbontása
rtin, rtout	MIDI	valós idejű rendszerüzenetek fogadása és küldése
seq	Adattárolás	szekvenszer
sustain	MIDI	note-off üzenetek visszatartása
sysexin, sxformat	MIDI	system exclusive üzenetek fogadása, küldése
touchin, touchout	MIDI	aftertouch érzékelés, küldés

F1/c melléklet: alapvető MSP objektumok

Objektum	Kategória	Leírás
*~, +~, -~, /~	DSP	aritmetikai művelet hangjelen (pl. *~: hangerő-módosítás)
adc~, dac~	I/O	hang be- és kimenet
allpass~, biquad~, buffir~, lores~, onepole~, reson~	Filter	mindent áteresztő, FIR, bufferalapú, aluláteresztő, egypólusú aluláteresztő és rezonáns sávszűrő
bitshift~	DSP	digitális jelfolyamot 1 bit értékkel eltol
buffer~, groove~, play~	DSP	hangminta tárolása és visszajátszása
click~	Jelgenerátor	impulzusgenerátor
clip~	DSP	audio-értéket vágással határok közé szorít
comb~, teeth~	Filter	comb filter
cycle~, phasor, rect~, tri~/triangle~, saw~	Jelgenerátor	színusz-, fűrészfog-, háromszög- és sávlimitált fűrész-jelgenerátor

degrade~, overdrive~, round~, trunc~	DSP	hangjel újramintavételezése ill. torzítása minőségromlással
delay~	DSP	késleltetés
fft~, ifft~	DSP	gyors fourier-transzformáció
freqshift~, gizmo~	DSP	hanghossz- és magasság változtatása fft-vel
ftom~, mtof~	Konverzió	frekvencia-hangjegy konverter
gate~, matrix~, selector~	Kontroll	hangjel útjának vezérlése: választás több kimenet ill. több bemenet között
meter~, levelmeter~, scope~, spectroscope~	Vizualizáció	hangerő, hullámforma és spektrogram megjelenítése
noise~, pink~, rand~	Jelgenerátor	fehér-, rózsazaj- és paraméterezhető zajgenerátor
normalize~, omx.comp~, omx.4band~, omx.5band~	DSP	normalizálás és dinamika-kompresszió
receive~, send~	Kontroll	hangjel továbbítása kábelek nélkül
sah~	DSP	sample and hold: közvetlen mintavételezés bufferbe
sfplay~, sfrecord~	DSP	hangállomány-lejátszás, -rögzítés
sig~, snapshot~	Konverzió	hangjel generálása törtszámból és fordítva
vst~	DSP	vst plug-in behívása
waveform~	Vizualizáció	buffer hullámforma-nézet és szerkesztési lehetőség

F1/d melléklet: alapvető Jitter objektumok

Objektum	Kategória	Leírás
*~, +~, -~, /~	DSP	aritmetikai művelet hangjelen (pl. *~: hangerő-módosítás)
adc~, dac~	I/O	hang be- és kimenet
allpass~, biquad~, buffir~, lores~, onepole~, reson~	Filter	mindent áteresztő, FIR, bufferalapú, aluláteresztő, egypólusú aluláteresztő és rezonáns sávszűrő