

Hangkezelés a Pd-vel

A Pd segítségével akár egy visszhangszűrő vagy valamilyen más hangeszköz vizuális létrehozására is képessé válunk.

A Pure Data (Pd) valós idejű vizuális programozói környezet hang- és egyéb multimédia-alkalmazások létrehozására. Segítségével foltokat (patch) hozhatunk létre, amelyek a hang- és képadatokon valamilyen műveletet hajtanak végre. Ezeket a foltokat a program vizuálisan ábrázolja: rajzolásal adjuk meg, hogy a jelek adatai merre haladjanak és mi történjen velük. A folyamat hasonlít a különböző részekből felépülő analóg szintetizátorok programozásához, és jól illeszkedik a hang- és videóalkalmazások programozásához. A jelek a rendszerbe befelé és abból kifelé haladnak, és a rendszeren való áthaladás közben megfelelő módon módosíthatjuk őket. Terjedelmi okokból ez a cikk csak a Pd kiforrottabb hangképességeit mutatja be. Ha később a képfeldolgozó bővítményeket (plugin) is ki szeretnénk próbálni, látni fogjuk, hogy az elvek nagyon hasonlóak a hangfeldolgozó részben megismertekhez.

Hangjelek és üzenetek

A Pd az adatok két alaptípusát különbözteti meg: az üzeneteket és a hangjeleket. Az üzenetek a MIDI hangkimenethez hasonlóan csak szórványosan jönnek létre, számokat vagy karakterláncokat tartalmazhatnak, és például olyan üzenetek átadására használatosak, mint az „állítsd a kimenet erősítését x értékre”. A hangjelek állandóan jelen vannak, a hang átvittele a DSP (Digital Signal Processor) minden egyes bekapcsolásakor megtörténik. A Pd belsejében a hangok 32 bites lebegőpontos számok formájában tárolódnak, ami azt jelenti, hogy a hagyományos analóg vagy digitális

hangfeldolgozással ellentétben a Pd jelei szinte bármilyen amplitúdóval bírhatnak. A feldolgozás során létrehozhatunk egy nagyon halk hangot az egyik szakaszban, amit egy másikban minőségromlás nélkül erősíthetünk fel. Természetesen, amikor a jeleket valamilyen eszközre engedjük, az értéknek a használható, -1 és 1 közé eső tartományba kell esnie, különben levágja (clipp) a hanghullám tetejét.

Az üzeneteket és hangjeleket az alább ismertetésre kerülő különféle dobozok (boxes) segítségével módosíthatjuk – ezeket összeillesztve az összekapcsolt dobozokat foltnak (patch) hívjuk.

A dobozok – dolgozzunk is egy kicsit!

Minden feladatot a dobozok végeznek el. A Pd a dobozok négy fő típusát különbözteti meg: objektum, üzenet, grafikus felület (GUI) és megjegyzés. Ezek a dobozok műveleteket hajtanak végre az üzeneteken és a hangon, lehetővé téve a felhasználói bemenetek megadását és az elvégzett tevékenységek feljegyzését. Az objektumdobozok további két típusra oszlanak: vezérlőobjektumokra és hullámvonalobjektumokra. A vezérlőobjektumok az üzenetekkel foglalkoznak, ezért a tevékenységük nem folyamatos. A hullámvonalobjektumok a hangjelekkel dolgoznak, emiatt állandóan működnek.

Az üzenetdobozok a tartalmukat a kimeneti kapujukra irányítják, amikor a felhasználó rájuk kattint vagy

	1. ábra Egy objektumdoboz
	2. ábra Egy üzenetdoboz
	3. ábra Egy GUI-doboz
	4. ábra Egy megjegyzésdoboz

amikor üzenet érkezik a bemenetükre. A GUI-dobozok egyszerűen csak hivatkoznak olyan dobozokra, amelyekkel kapcsolatba léphetünk, ilyen a bal oldalon lévő számdoboz is. Végül a megjegyzések teszik lehetővé szövegek bevitelét a foltba, más hatásuk nincs.

A Pd elindítása és az első foltunk létrehozása

Feltéve, hogy már túl vagyunk a Pd fordításán és telepítésén, meg kellene próbálnunk elindítani. Először is bizonyosodjunk meg róla, hogy beállítottuk a Pd futtatható fájljának setuid bitjét, és hogy a fájl tulajdonosa a rendszergazda. Bár biztonsági kockázatot jelenthet, ez feltétlenül szükséges a valós idejű ütemezéshez, ha a rendszergazdán kívül más felhasználóként is futtatni szeretnénk a Pd-t. Ha ezt nem tesszük és nem aktiváljuk a valós idejű ütemezést, akkor kattanásokat és pukkanásokat fogunk hallani minden olyan esetben, amikor egy másik folyamat – akár az X-kiszolgáló – bármilyen tevékenységgel próbálkozik.

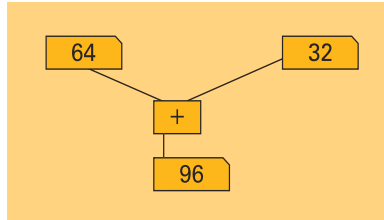


1. kép A főablak

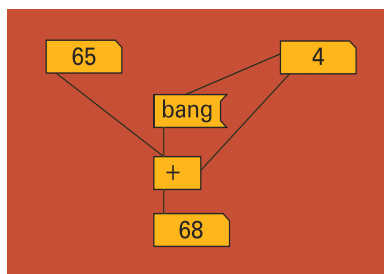
A Pd futtatásánál használjuk a `-rt` kapcsolót és emellett azokat a beállításokat, amelyeknek még szükségét érezzük. Javasolom a `-verbose` kapcsolót is, mivel a Pd amúgy nem túl bőbeszédű, és ezzel a beállítással hasznos ismeretekhez juthatunk. Ezután egy ahhoz hasonló ablakot kell látnunk, amelyet az 1. kép is mutat. Az **IN** és **OUT** feliratú dobozok a be- és kimenet hangszint csúcserőértékeit mérik, és a **peak meters** (csúcscintmérés) jelölőnégyzetre kattintva engedélyezhetők. Ha valamelyik túllépi a megengedett szintet, a rendszer levágja, és a megfelelő CLIP-doboz piros színű lesz. A **DIO errors** gomb abban az esetben kezd villogni, ha a bemenetben vagy kimenetben valamilyen összehangolási hiba lép fel. A gombra kattintva a legutóbbi hibák listáját jeleníthetjük meg. A **compute audio** (hang számítása) jelölőnégyzet a hangfeldolgozást kapcsolja be, illetve ki. Először is hozzunk létre egy új, üres foltot, amelyen dolgozhatunk (**File > New**) – ebből egy olyan egyszerű foltot fogunk készíteni, amelyik a „Szia Világ!” szöveget írja ki a szabványos kimenetre. Szükségünk van tehát egy üzenetdobozra, amelyben a „Szia Világ!” szöveget tároljuk, és egy objektumdobozra a nyomtatás végrehajtásához. Mindkettő a **Put** (elhelyezés) menüből hozható létre. Gyorsbillentyűk használatára is lehetőségünk nyílik: a **CTRL-1** az objektumdobozt, a **CTRL-2** pedig az üzenetdobozt helyezi el. Amikor ezzel készen vagyunk, kattintsunk a dobozra és gépeljük be a megfelelő szöveget, majd kapcsoljuk az üzenetdoboz alján lévő kimenetet az objektumdoboz tetején látható bemenethez. Foltunknak a 5. ábrán láthatóhoz hasonlóan kell festenie. A **CTRL-E** billentyűkombinációval lépünk ki a szerkesztőmódból. Próbáljunk rákattintani a „Szia Világ!” üzenetdobozra; ha minden rendben van,



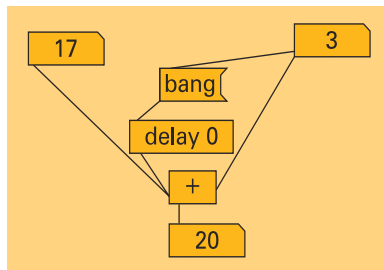
5. ábra A klasszikus „Szia Világ!” példa



6. ábra Két szám összeadása



7. ábra A forró és hideg bemenet kezelésének naiv módszere



8. ábra

A forró és hideg bemenetek helyes kezelése

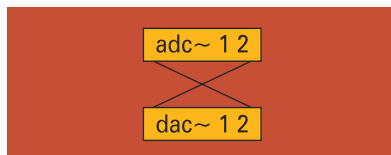
akkor az üzenetnek meg kell jelennie abban a terminálablakban, ahonnan a Pd-t elindítottuk.

Az üzenetek használata

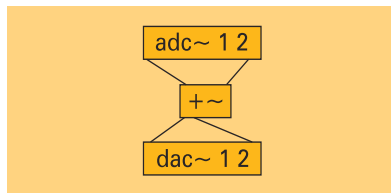
Most próbáljunk ki egy bonyolultabb példát: készítsünk egy olyan foltot, amely két számot ad össze és megjeleníti az eredményt. Ehhez hozzunk létre egy olyan foltot, amelyet a 6. ábrán látunk, két számdobozzal a tetején, középen egy objektumdobozzal, és lent ismét egy számdobozzal. Ezután lépünk ki a szerkesztésből (ismét **CTRL-E**) és próbáljuk megváltoztatni a fenti számokat. Ezt rákattintással és az

új érték beírásával, vagy kattintással és az egér húzásával végezhetjük el: felfelé mozogva növekszik az érték, lefelé pedig csökken. Valószínűleg észrevettük, hogy a bal oldali szám megváltozásakor a lenti eredmény azonnal megváltozik, a jobb oldali érték megváltoztatásakor azonban nem történik semmi. Vajon mi lehet ennek az oka? A Pd csaknem minden objektumára igaz, hogy a bal szélén lévő bemeneti csatlakozót forrón tartja, azaz ennek bármilyen változása azonnal megjelenik a kimeneten. A többi bemenet hideg, értékük megváltozása nem okoz változást a kimeneten. Az új érték mindaddig egyszerűen csak tárolódik, amíg a forró bemenet ki nem váltja a számítás elvégzését – ekkor kerül az új érték felhasználásra. Hogyan érhetnénk el, hogy a hideg bemenet változása is megváltoztassa a kimenetet? Az egyik módszer egy üzenetdoboz beszurása. A 7. ábrán a jobb oldali számdoboz kimenetét egy üzenetdoboz bemenetére kötöttem, majd ennek az üzenetdoboznak a kimenetét az alatta lévő összeadó objektumdoboz forró bementi csatlakozójára. Amikor egy üzenetdoboz bemenetére bármilyen üzenet érkezik, a benne tárolt tartalmat új üzenetként a kimenetére küldi ki. Így amikor a jobb oldali számdoboz tartalma megváltozik, az egy üzenetet küld a bang-üzenetdoboznak, amely egy bang-üzenetet (bang message) küld az összeadó objektumdobozra. A bang-üzenetek jelentése az, hogy „Csinálj valamit!”, vagyis ha egy objektumdoboz ilyen üzenetet kap a forró bemenetére, azonnal végrehajtja a kért számítást. Most ezt a viselkedést arra használjuk fel, hogy az összeadó objektumdobozunk olyan módon működjön, mintha két forró bemeneti csatlakozóval rendelkezne. De várjunk csak! Ez nem azt jelenti, hogy a bang-üzenetnek a szám után kell megérkeznie? Ha nem így történik, a folt nem fog működni, igaz? Nos, igen: attól függően, hogy a kapcsolatokat milyen sorrendben hoztuk létre, tapasztalhattuk, hogy nem működik. És csakugyan, a 7. ábrán látható számok összeadása e miatt a hiba miatt nem pontos, vagyis szükségünk van valamilyen módszerre, amellyel elérhetjük, hogy a bang-üzenet a szám után érkezzen meg. Ennek az az

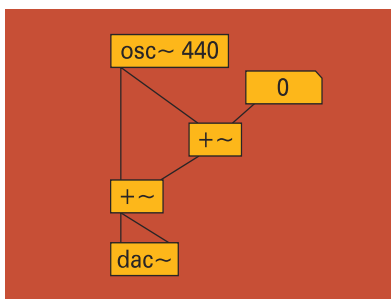
© Kiskapu Kft. Minden jog fenntartva



9. ábra A jobb és bal oldal felcserélése egy sztereójelben



10. ábra A sztereójel monóvá alakítása



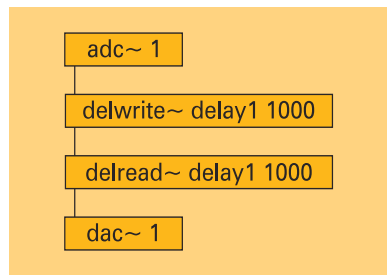
11. ábra Két jel kioltása inverz jel használatával

egyszerű módja, hogy egy késleltetést iktatunk be, mint az a 8. ábrán is látható. Érdekes módon a 0 késleltetés is működik. Az üzenet egy DSP-ciklusnyi késleltetést szenved, ilyen módon biztosított, hogy a bang-üzenet érkezik be másodikként.

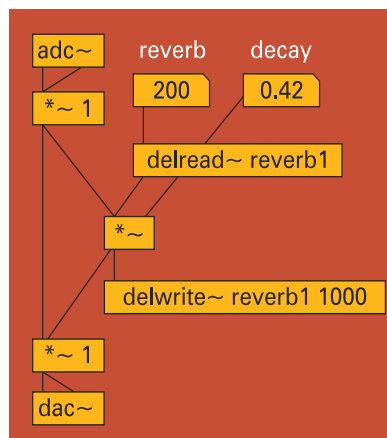
Hangfeldolgozás

A hangfeldolgozás legalapvetőbb függvényei a bemenet és a kimenet. Az `adc~` hullámvonal-objektum jelképezi az analóg–digitális átalakítót, amely az első feladatot hajtja végre, a `dac~` hullámvonal-objektum pedig a digitális–analóg átalakítót, ennek szerepe a kimenet előállítás. Alapértelmezésben mindkettő az első két csatornán működik. Ha ezt meg szeretnénk változtatni – például olyan többcsatornás hangkártyánk van, mint amilyen a Hasmmerfall HDSP –, értékként megadhatjuk a csatornák számait, ezáltal a megfelelő csatornák hozzárendelődnek megfelelő bemeneti és kimeneti csatlakozóikhoz. A 9. ábrán látható egy egyszerű példa, amelyben a sztereóbemenet csatornái

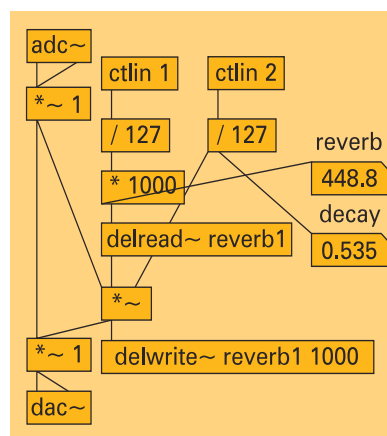
felcserélődnek és a kimenetre vannak irányítva. Mivel a sztereóbemenet az alapértelmezett, szükségtelen megadni a csatornák számozását, de szemléltetés céljából most feltüntettem. A hangok adatai egy bizonyos mintavételezéssel érkező számsorozatok, így ezeken alkalmazhatók a



12. ábra Egy késleltetési vonal



13. ábra Egy egyszerű visszhangszűrő



14. ábra

A MIDI használata egy folt vezérlésére

matematikai műveletjelek (operator). Például a 10. ábrán látható módon a sztereójelből monó hangot állítunk elő azáltal, hogy a jobb és bal csatornát

összeadjuk. Egy másik hasznos műveletjel a szorzás (`*~`), amit erősítőként alkalmazhatunk. Ne feledjük azonban, hogyha a hangot valamilyen eszközzre bocsátjuk, a jeleket levágja, amennyiben az értékük nem esik -1 és 1 közé.

A 11. ábrán egy összetettebb példát szemléltet. Érdekes lehet lehalkítani a berendezésünket, mert az eredmény elég hangos. A fenti `osc~` objektum egy szinuszel-generátor, ami ebben az esetben 440 Hz-en fut. Az előállított jel két részre válik: a bal fele közvetlenül az összeadóra jut, míg a jobb oldali rész először egy szorzáson megy keresztül.

Most próbáljuk meg a -1 értéket beírni a számdobozba – a hang elhallgat. Vajon mi lehet az oka? Ha még vissza tudunk emlékezni a középiskolában a hullámok fizikájáról tanultakra, eszünkbe juthat, hogy minden hullám kioltatható egy másikkal. Ebben az esetben a -1 az eredeti jel tökéletes inverzét állítja elő, így ahol az eredeti jel értéke 1, az inverzé -1. Ha ezt a két jelet összeadjuk, nullát kapunk, ami a csendnek felel meg. Kipróbálhatjuk azt is, hogy a SHIFT billentyűt lenyomva tartjuk és az egérkurzort a számdobozon húzzuk; a számok elég lassan változnak ahhoz, hogy halljuk elhallkultni és megszűnni a hangot, ahogy elérjük a -1-et.

Egy egyszerű visszhangszűrő létrehozása

Építsük össze ezeket az alkotóelemeket és próbáljunk meg kihozni belőlük valami használható dolgot: készítsünk egy egyszerű visszhangszűrőt (Reverb Filter) a fenti módszerek használatával, hozzávéve még egy újat, a késleltetővonal (Delay Line). Ha már van némi tapasztalatunk a hanghatások (sound effects) tervezésében, valószínűleg tudjuk, mi az a késleltetővonal, ha nincs, képzeljük egy hangok tárolására alkalmas átmeneti tárnak. A hang, amely a késleltetővonal bemeneti csatlakozójára kerül, egy meghatározott ideig tárolódik, mielőtt megjelenne a kimeneten. Az analóg korszak minden bonyolult módszerében fellelhető volt a késleltetés valamilyen megvalósítása, ezek némelyike különleges rugókat és szalaghurkokat használt.

Szerencsénkre mi már digitális jelekkel foglalkozunk, itt a késleltetővonal létrehozásához nincs szükség többre, mint egy FIFO tárolóra. A visszhanghatás egy olyan természetes jelenség, ami a hangnak a környezetében lévő felületekről történő visszaverődésének következménye. A nagyobb távolság miatt a visszaverődött hang egy kicsivel az eredeti hang után szólal meg. Ezek a visszaverődött hangok saját maguk is visszaverődnek. A legtöbb környezet rendelkezik valamilyen fokú visszhanghatással, sőt ennek a hatásnak egy pontosan szabályozott mértéke a koncerttermek igen meghatározó tulajdonsága. A nem túl erős akusztikával rendelkező termekben a visszhanghatás hiánya vagy csekély volta miatt az az érzésünk, mintha a hangot egy nem körülhatárolt térbe eregetnénk, mintha az nem verődne vissza. Ennek a szűrőnek a lelke a késleltetés. A 12. ábrán a hangbemenet 1 másodperces (1000 milliszekundum) késleltetését létrehozó késleltetővonalat látunk. A Pd-ben a késleltetővonalak névvel rendelkeznek, ami annyit jelent, hogy használatukhoz két különböző objektum szükséges: egy író és egy olvasó. Az író, a `delwrite~`, két értékkel bír: az első a késleltetővonal neve, a második pedig a használni kívánt legnagyobb késleltetés ideje ezredmásodpercekben megadva. Egy késleltetővonal csak egy íróval rendelkezhet. A másik objektum az olvasó, a `delread~`, amelynek szintén két értéke van: a késleltetővonal neve és a kívánt késleltetés mértéke. Az íróval ellentétben tetszőleges számú olvasónk lehet tetszőleges késleltetésekkel. Létezik az olvasó egy további típusa, a változó késleltetésű objektum (`vd~`), amely egy hangjel hatására változtatja a késleltetés mértékét, de ez már túlmutat cikkem keretein. Összegezve: a visszhangszűrőnek szüksége van egy közvetlen jeltömbre és egy vagy több késleltetett jelre, a visszaverődésekre. A visszaverődések rekurzívak, tehát a rendszerben valamilyen visszacsatolásnak is lennie kell. Végül, bár a bemeneteink és kimeneteink sztereójelet kezelnek, helytakarékoságból most csak egy egyszerű monó visszhangszűrőt valósítunk meg. Ehhez össze kell keverni a bejövő jobb és bal csatornát, majd a

másik felén kettéosztani. A 13. ábrán láthatunk egy ilyen szűrőt. Ha a bemenettől követjük a jel útját, az első dolog, amit észre kell vennünk, a `*~ 1` objektum. Ennek az a célja, hogy egy kicsit áttekinthetőbbé tegye a foltot. Ez a jelet eggyel szorozza, ami önmagában felesleges, viszont bemenetet ad mind a jobb, mind a bal csatorna számára, és egyetlen csatornává keveri össze a jeleket. A folt alján még egy hasonló elrendezést láthatunk. A visszhang jele ezután egy újabb szorzón megy keresztül, amely csökkenti a visszaverődés szintjét, mielőtt egy újabb visszhangciklusba kerülne. A `delwrite~` és a `delread~` képezik a visszaverődési ciklust. Végül a szorzó kimenete a `dac~`-ra kerül vissza.

MIDI-vezérlés hozzáadása

Először is győződjünk meg arról, hogy a használni kívánt MIDI eszköz különösen megfelelően működik, ha az Evolution UC-16-hoz hasonló vezérlőeszközzel van szó. Egy egyszerű módszer annak ellenőrzésére, hogy a Pd fogadja-e a MIDI-ről érkező jeleket: válasszuk ki a bármelyik ablak jobb felső sarkában megtalálható **Help** (súgó) menü **Test Audio and MIDI** (a hang és MIDI kipróbálása) menüpontját. Nyissuk meg, és állítsunk elő néhány MIDI-eseményt. Ha a vezérlőegységünk működik és vezérlőjeleket ad ki, akkor – ellentétben a megjegyzéseseményekkel – az ablak alján látható `ctlin` (control in, vezérlőjelbemenet) objektumok alatti számoknak változniuk kell. Előfordulhat, hogy ehhez játszaniuk kell egy kicsit a Pd `-midi` parancssori kapcsolójával, és ne felejtjük elolvasni a `-verbose` kapcsolóval indított Pd elindulásakor megjelenő hibaüzeneteket. Ha mindez működik, próbáljuk ki a 14. ábrán látható foltot, amely a 13. ábrán látható visszhangra hasonlít, azzal a különbséggel, hogy a visszaverődést és a késleltetést most a MIDI és nem a GUI vezérli. Ez a váltás a `ctlin` objektum segítségével valósítható meg, amelynek ebben a példában egyetlen értéke van: a használni kívánt vezérlő száma. Kimenete a vezérlőről érkező szerkezet nélküli érték. Ha UC-16-ot vagy ahhoz hasonló vezérlőt használunk, ez az érték 0 és 127 közé eshet. A többi objektumnak egy kis matematikai

átalakítást kell végeznie ezen az értéken ahhoz, hogy a kívánt tartományba essen. Visszhang esetén ez a tartomány 0-tól 1000-ig terjed, ezért először 127-tel osztunk, hogy 0 és 1 közé eső számot kapjunk, majd ezt szorozzuk 1000-rel, hogy a kívánt értékhez jussunk. A csillapítás számára a 0 és 1 közé eső érték megfelelő, ezért már csak osztanunk kell. Jó gyakorlat jelent vizuális visszajelzést adni a hatás beállításának értékeiről, ezért ez a folt mind a visszaverődési, mind pedig a csillapítási értékekről a megfelelően jelölt számdobozokba küld egy-egy másolatot. Meg is volnánk: van egy teljesen számíttógépesített, MIDI-vel vezérelt hanghatásunk, amelyet csupán néhány Pd-objektum összekapcsolásával hoztunk létre. És ez csak a kezdet a lehetőségeknek. Van egy MIDI billentyűzetünk? Használhatjuk vezérlőként és írhatunk egy olyan foltot, amelyik meg is szólaltatja a hangokat; vagy egy olyan foltot is létrehozhatunk, amely a MIDI billentyűzetet vezérli. Akár teljes, újraprogramozható hanghatásdobozt (effects box) is létrehozhatunk. Nem kell megelégednünk a már kész Pd-objektumokkal sem, írhatunk saját objektumokat C nyelven vagy akár magában a Pd-ben. Ha a Linux és a Pd a társunk, akkor az egyedüli korlát a saját tudásunk és a processzorunk sebessége.

Linux Journal 2003, 116. szám



Peter Todd 14 éves kora óta használja a Linuxot. Másodállásban egy kis Linux alapú hangstúdió vezető technikus. Amikor éppen nem dolgozik, akkor a Wexford Főiskolát látogatja, ahol jelenleg kerámia- és tervezőgrafika szakon tanul.

KAPCSOLÓDÓ CÍMEK

A cikk minden példája elérhető a Linux Journal FTP-oldalán az <ftp://ftp.ssc.com/pub/lj/listings/issue116/7062.tgz> címen. A reverb, delay rejtelmeiről a <http://www.emilrulez.hu/delay.html> oldalon olvashatunk.