*As published in Sound On Sound (UK) and Music & Computers (US) magazines*, **1997**

## Computers Come Alive:

# The Making of "King Frank"

### A Live-Performance Piece for Six Players, Four Samplers, One Computer, and One Late, Great Composer

By Paul D. Lehrman

Computers are great for helping out with home and office tasks, but why would anyone ever dream of hauling one up onstage as a bandmate? The waking reality is that desktop computers offer new possibilities when it comes to live music. Thanks to MIDI, sequencers, and sampling technology, a keypress or breath of air can translate into a whole universe of musical (or non-musical) sounds. What might look like one kind of instrument onstage might sound like something completely different, or not even an instrument at all. The identity of an instrument can change from moment to moment.

The concept of sharing the stage with a computer raises many questions; the first is often, "Who’s in charge here?" How much control of the performance do you want to keep in your own hands, and how much do you want to give to the machine? At one end of the scale, the computer does all the work and the performer is basically irrelevant, or at least ignored by the machine. At the other end, the performer is in control all the time, and the computer does nothing unless the performer tells it to.

Performing with computers doesn't have to be an either/or situation when it comes to who's in the driver's seat.

"*King Frank*," a piece I conceived and composed recently for myself and five other players, lies somewhere between these extremes. In this piece, the computer and musicians play off of each other, and the performance control switches back and forth, sometimes in ways that the audience can see, and sometimes not. It’s in this middle ground that the most exciting possibilities for computer-assisted performance lie.

Not too many years ago, a project of this complexity and flexibility would have required custom programming on a large computer, using state-of-the-art sampling technology, and special "gesture" MIDI controllers. But I created King Frank with off-the-shelf hardware and software, at the hub of which was a five-year-old Macintosh *Quadra 650*.

"*King Frank*" is a tribute to the late Frank Zappa that borrows heavily from Zappa’s body of recorded works. I developed it as a term project for the students in my advanced Computer

Applications in Music seminar, in the Sound Recording Technology program at the University of Massachusetts Lowell. We performed the piece as part of a concert consisting of student MIDI and digital audio compositions.

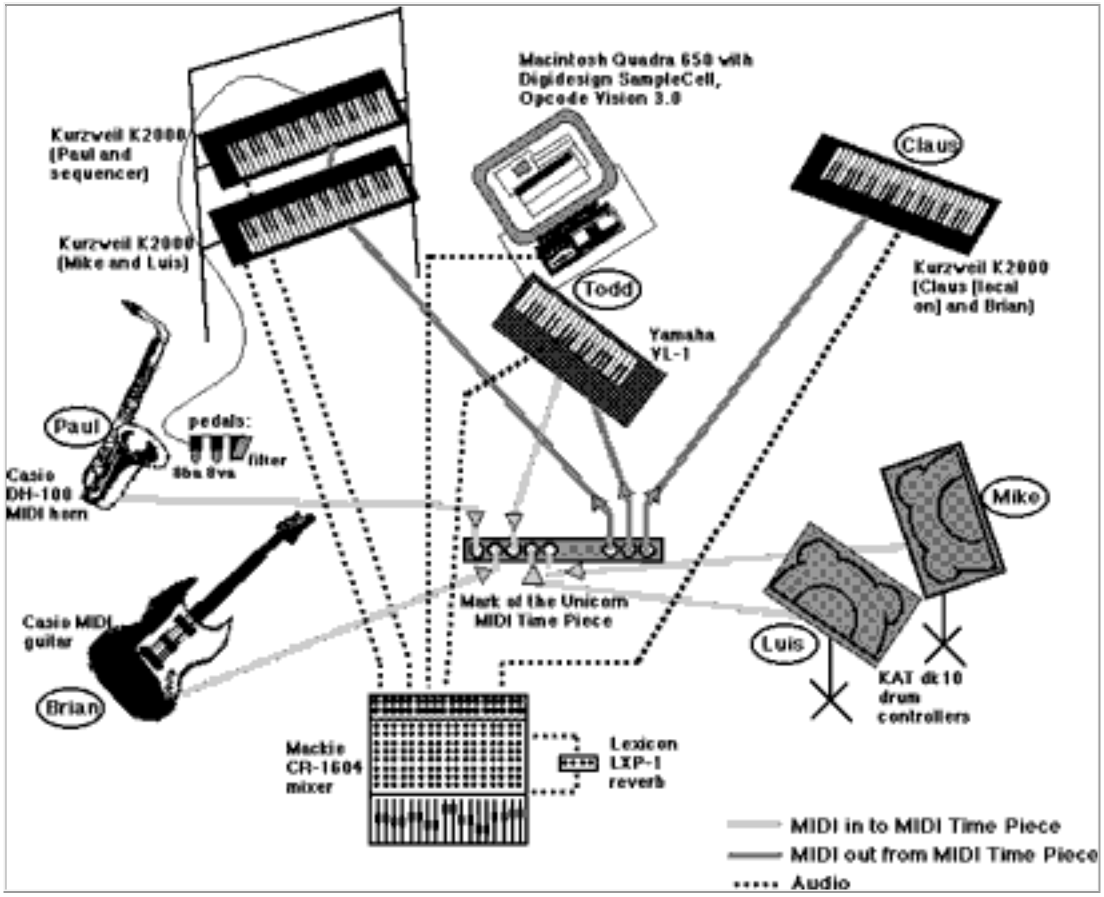
At the heart of "*King Frank*" is "*King Kong*," a fast, 3/4 modal jazz tune originally on the Mothers Of Invention’s Uncle Meat. A lot of "*King Frank*" borrowed from standard jazz structure: The piece started with an introduction and a statement of the tune, followed by each of us "blowing" a solo on the tune — the last solo being, of course, drums. The six performers used various MIDI controllers — two keyboards, guitar, horn, and two drum pads — to play the melody, fill in harmonies, and solo while the MIDI sequencer running on the computer provided basic rhythm tracks.

Instead of playing MIDI synthesizers, most of us played samplers so that we could switch between normal instrument sounds and non-instrumental samples instantaneously. The idea was for the tune to evolve as a collage of harmonic material and what can only be described as audio mayhem. At various points in the piece, the music would stop dead and the audience would hear sampled phrases from a Zappa record such as, "The way I see it Barry, this could be a very dynamite show!" We called these interludes "samplus interruptus." Besides being funny, they kept both the performers and the audience on their toes.

During a free-form section that followed our solos, we bombarded the listeners with odd phrases, noises, sound effects, and loops from Zappa records until we reached a glorious cacophony. Following this wall of sound, we went into a "trading 8’s" section — exchanging eight-bar solos over the tune’s chord changes. But from the audience’s standpoint, it seemed that we had emerged from the preceding chaos with all the wrong sounds! The drummer was playing a guitar, the guy at the keyboard was playing the flute, and the sax player was playing, of all things, a sitar! Indeed, we were not just trading licks; for the moment we had swapped instruments, and the sounds each of us played in our little solos bore no physical resemblance to the instruments we were actually playing.

After another interruptus from Frank, we launched into the head (main melody) one final time, with all the instrument sounds apparently back where they belonged. The performance came off smothly, but it didn’t just work out that way. Behind our performance was a good deal of planning, creative routing of MIDI signals, and a twisted set of samples to top it all off.

## Share and Share Alike



Probably the most involved aspect of our performance was how we shared the samplers and MIDI controllers with the computer and each other. Each player had one MIDI input device, but not necessarily a tone generator of their own.

Three Kurzweil K2000S sampling keyboards and a Digidesign SampleCell card, which lives inside the Mac, were called into service as sound sources. The K2000S samplers were configured so that each of them functioned as two distinct instruments: The keyboard that Claus was playing had one MIDI channel devoted to his sounds, while a second channel produced the sounds that Brian was playing from his guitar. Similarly, the two drummers shared another K2000, each with his own MIDI channel and his own audio output pair. The third K2000 was shared by my MIDI horn and the sequencer itself, which played the interruptus samples and backing tracks. Todd triggered sounds from the SampleCell card by routing MIDI signals from his Yamaha VL-1 keyboard to the card using the sequencer's Thru function (which allows incoming MIDI signals to be passed through the computer to a receiving MIDI device).

## Ample Samples

We used two different types of instruments to play samples for "King Frank." The first was a Kurzweil *K2000S* MIDI synthesizer with sampling capabilities. The other instrument was a Digidesign *SampleCell*, which is a card that plugs into a slot inside of a Macintosh (both PCI and NuBus versions are available). You can't play its sounds directly with a keyboard. Instead, you need to route MIDI information to it using the computer's operating system. OMS, the MIDI operating system extension from Opcode, includes a SampleCell driver, so that MIDI programs like Vision (and connected MIDI input devices) can trigger and control the sounds in SampleCell.

Creating the various Zappa-sample patches was a process that began with going through some of my rather vast collection of Zappa CDs on Rykodisc and Zappa’s own Barking Pumpkin label, in my home studio. I had hoped to use the the internal CD-ROM drive on my Macintosh clone to gather the samples, using the "digital audio extraction" feature built into *QuickTime*, which allows you to import recordings from an audio CD directly onto your hard disk, with your choice of sample rates, word lengths, and number of channels. I quickly found, however, that this was very cumbersome. The extraction process provides no easy way to audibly cue the CDs—you have to set up the in and out points by specifying timing numbers. If you want to hear the CD, you have to use CD Remote or another software controller, most of which are clumsy, inaccurate, provide no high-speed cueing, and often have the annoying habit of jumping to the beginning of a track when you’re trying to cue backwards.

## What are those samples???

Some of you are no doubt asking how we could blatantly sample Zappa’s work without worrying about the copyright police. The answer? "Fair use" — the principle that allows anyone to use copyrighted works for certain non-profit purposes (educational use being one of the most common) without getting permission or paying royalties.

In addition to the ones mentioned in the main story and the key-mapping illustration, here are some of the other samples we used in *King Frank*, and their source. All CDs are on the Rykodisc label.

**Absolutely Free**
*Uncle Bernie’s Farm*, "There’s a bomb to blow your daddy up"
*Brown Shoes Don’t Make It*, "Be a joik and go to woik"

**Apostrophe**
*Don’t Eat the Yellow Snow*, "Don’t you go where the huskies go"

**Chunga’s Revenge**
*Would You Go All the Way?,* "Lift up your dress"

**Freak Out**
*It Can’t Happen Here*, "AC/DC" (looped)
*Help I’m a Rock*, "America’s wonderful/wonderful" (looped), "It’s a drag being a cop"
*Wowie Zowie*, "I don’t even care if you shave your legs"

**Meets the Mothers of Prevention**
*Porn Wars*, "It’s outrageous filth"

**Overnite Sensation**

Fortunately, I have a JVC CD audio player that happens to be one of the few consumer players ever produced with an S/PDIF output. I connected that to the digital input on my Digidesign *Pro Tools* Audio Interface, and launched *Sound Designer II*. I put the software in "Monitor" mode to listen to the CDs, and when I encountered a sample I liked, I simply backed up the CD the requisite number of seconds (using a wireless remote, so I didn't even have to sit up) and put the software into Record. *Sound Designer* has an advantage over some audio recording programs that when you record a signal in mono, it combines the two channels, rather than just leaving one out—since there was no particular reason to record the samples in stereo, I used this feature for all the samples.

*Montana*, "a pair of heavy-duty zircon-encrusted tweezers", "yippie-i-o-tie-yay"

**We're Only In It for the Money**
*Who Needs the Peace Corps*, "phony hippies", "Flower power sucks/sucks" (looped, of course)
*Nasal Retentive Calliope Music*, "a little nostalgia for the old folks"

**You Can't Do That On Stage Anymore, vol. 1**
*The Groupie Routine*, "with a bullet", "That's me"

One feature *Sound Designer* no longer has, however, is a way to send files to an external sampler, and so after I collected about 150 samples, I broke them up into sets corresponding to the five players who would be using them, and brought them into BIAS' *Peak*. There I did some quick trimming and normalizing of the files (things that Peak is much faster at than Sound Designer, due to its working with both RAM and disk files, as opposed to the older program's relying entirely on disk operations), and sent them over SMDI—the SCSI-based file exchange program used by most sampler manufacturers—to my *K2000*, through the fairly complex SCSI chain in my studio. The latest version of Peak (1.5) allows batch normalizing, which saved a lot of time, and even better, it allows SMDI transfers to be batched, which meant I could set up the operation, and walk away for 20 minutes, and when I came back all the samples were nicely lined up inside the *K2000*.
I then created keyboard maps for each player in the *K2000* to hold the appropriate samples, and created patches from the keymaps. For the two drummers, I set up the keymaps to match the General MIDI note map built into the dk10s, and also specified that would be ignored—the whole sample would play whenever they hit a pad, regardless of its length. For the others I arranged the samples so that each one covered a range of two or three half-steps, so that they could be played several times and each time they would sound slightly different, since they would be pitched differently. I saved the finished patches, with the samples, to the internal drive on my *K2000*, and backed it up to a Zip cartridge, formatted for the *K2000*.

Once the samples were inside my sampler, I created keymaps — assignments of specific samples to individual MIDI notes. For example, the *K2000* program Brian ended up with consisted mainly of guitar loops assigned to MIDI notes that he could play with his guitar controller. Two samples were assigned to each string, and the top string got five additional samples, going up to high A.

Since *SampleCell* uses standard AIFF files for its samples, I was able to leave the samples Todd would end up using on the hard drive for him to arrange into keymaps with SampleCell's editing software. His set included tasty tidbits like, "Why does it hurt when I pee?" from **Joe's Garage**.

I brought my Zip drive and cartridges up to school, and transferred the sample and program files to the hard disks in the MIDI studio, one of which is used for Sound Designer and Pro Tools files, and the other is dedicated to the Kurzweil *K2000*, although they are all on the same SCSI chain. Normally there is only one *K2000* in the studio; the other one lives in our 24-track room, but I stole that one for this project. After loading one *K2000* with samples, the SCSI connector had to be switched by hand to the other *K2000*, so that its samples could be loaded. We had to do this each time we rehearsed, and before the performance. You can be sure that after we got everything set up for the performance, we guarded the main power switch for the room closely!

## Designing the sounds

The Casio *DH-100* MIDI Horn, which I was playing, is a curious beast. Like the Mattel *Power Glove*, which had a very short commercial lifespan but now is in great demand by experimenters as a real-time virtual reality controller, the *DH-100* never really caught on, and Casio dumped it. A small company in Wisconsin, however, picked up Casio's inventory and still sells the unit at a bargain-basement price. It looks pretty silly, and is built rather flimsily, but once you get to know it, it can be a genuinely useful MIDI controller.

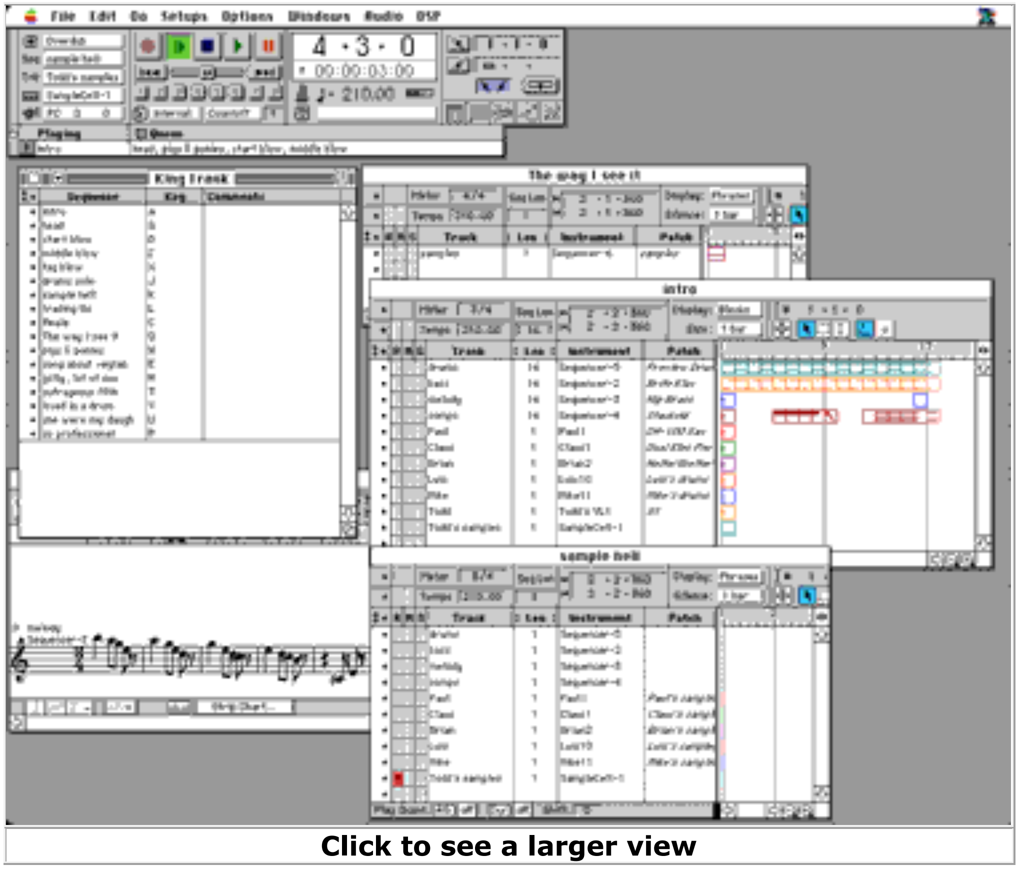
It generates only three types of commands: notes, aftertouch, and portamento. The note numbers are determined by how you place your fingers on the keys, and there are two fingering schemes available: one that closely resembles a normal saxophone, and gives you a 2-1/2 octave range, and a weird "binary" sort of fingering that, if it were at all playable, which it isn't, would yield a four-octave range.

You blow into a mouthpiece, behind which is a breath sensor. When the pressure on the breath sensor passes approximately the half-maximum point, a note-on is generated, immediately followed by aftertouch, which then follows changes in the breath pressure. Therefore, you must be blowing to play a note (the breath sensor is defeatable, but that sort of negates the whole value of the thing, doesn't it?), and the initial value of the aftertouch under a note will always be at least 64. The note continues, and aftertouch continues to be generated, until you stop blowing completely—if you pass below the half-point threshold, you still get sound. So, while it's not terribly easy to sneak in a note quietly, it is easy to do a long fade on one. Portamento is turned on with a switch by the left index finger.

I created a patch on the *K2000* that used the Tenor Sax patch in ROM as a take-off point, assigning aftertouch to the patch's volume, as well as to a filter that provided a little midrange boost as I blew harder. I also assigned foot controller (MIDI controller #4) as a second source for the same filter, configuring it to provide a lot more boost, and also to alter the filter's resonant frequency—thus achieving that horrid Ian Underwood late-'60s electric-sax-with-wah-wah sound that Zappa was so fond of. I put the patch into Mono mode, and gave it a portamento range of 70 keys/second, so that I could do little slides or huge swoops, toggling them with the horn's portamento switch.

In addition I used two foot switches to control the pitch, one of which raised it by an octave (1200 cents) and the other lowered it by an octave. This gave my MIDI horn a range of over four octaves. Interestingly enough, I found that lowering the Tenor Sax ROM samples resulted in a very convincing baritone sax—gutsier and sloppier than using a real bari sax sample.

## Queue the Cues



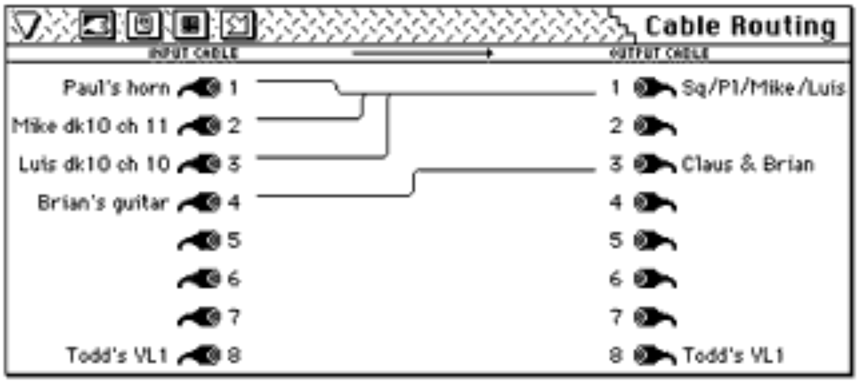
Click to see a larger view

We were running Opcode’s *Vision 3.0.1* MIDI sequencing software on a Macintosh *Quadra 650* for the entire piece. *Vision* allows multiple sequences to be loaded at the same time and played in any order by cueing them from the Macintosh keyboard — this was part of Todd’s job. We had separate sequences for the 16-bar introduction, head, instrumental solos, and each of the special sections. *Vision*’s cue functions gave us an edge: If you cue up a second sequence while another is playing, the second one will wait for the first to finish, then start immediately. So when sequences needed to be strung together without interruption, as when the introduction ended and the head began, Todd could pre-cue the second sequence, and it would start automatically at exactly the right time.

But there were times when we wanted the computer to follow us, like after the samplus interruptus sequences. We made those sequences very short — one eighth-note in a single 1/8 measure. So while Todd would pre-cue a sample sequence to follow the head or a solo, he wouldn’t pre-cue sequences that followed the interruption segments. When a musical section ended, the sequencer would play a sampled phrase by triggering a single note, then immediately stop and wait. When the sample finished, Todd would nod his head, and simultaneously hit a Macintosh key to start the next sequence, thereby cueing the downbeat for both the computer and the performers. (see Figure 3) One important parameter in the samplers had to be modified to make this work: The "Ignore Release" switch in the K2000 program that contained the interruptus samples had to be turned on. This meant that the sample would play through to its end, regardless of when the MIDI note that had triggered it stopped.

For our solo section, we constructed three 16-bar sequences to play underneath the solos: "start," "middle," and "tag." Each of these contained additional instrumental tracks that provided distinct accents that acted as aural references so we could hear where the computer was at in the sequence. The start and tag sequences were fixed in length, but the middle sequence was set to loop indefinitely. This allowed each solo to be a different length. It was the soloists' job to somehow communicate with Todd using facial gestures. When a soloist felt it was time to wrap it up, Todd would cue up the tag sequence. When the currently looping middle sequence reached its end, the tag sequence would kick in. Once the tag ended, Todd would make sure another interruptus sequence was already cued up, so it would sound as if the music was being rudely cut off by another non-sequitur.

## Signal Routing: What the Audience Didn't See



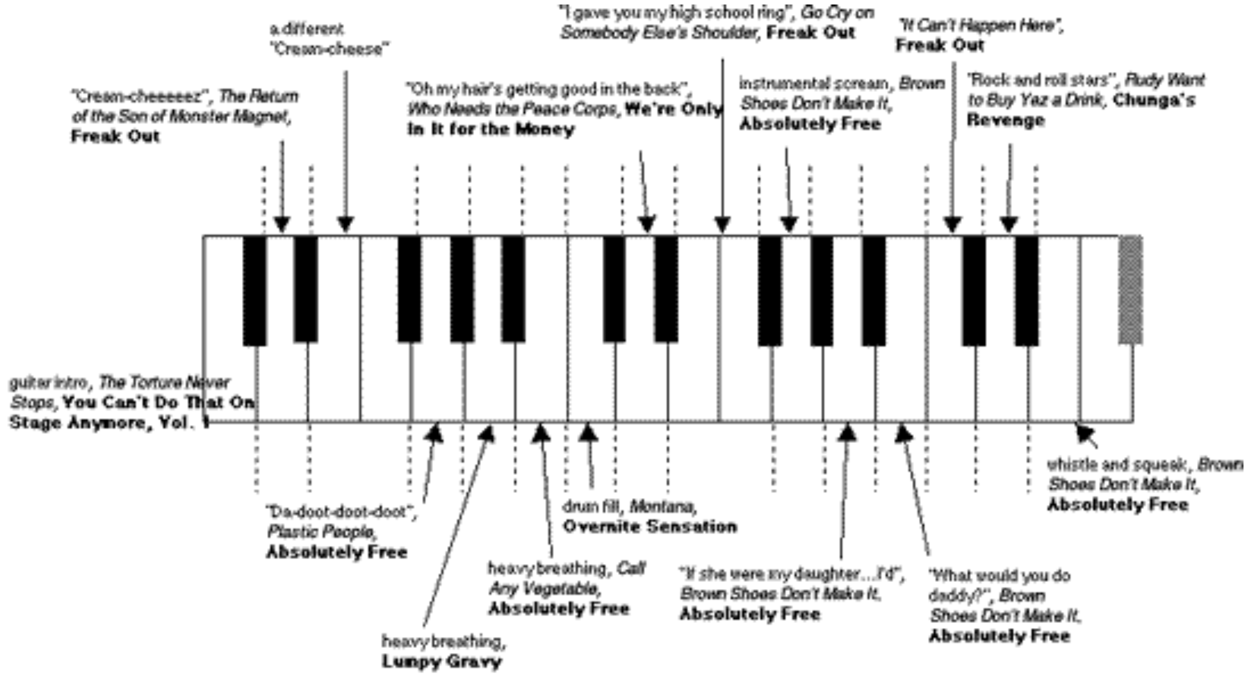
One of the most difficult jobs in the creation of "King Frank" was working out the MIDI routing. We were faced with an interesting problem: All of the tone-generating instruments needed to be under the control of both the musician and the computer (which was playing sequences and sending program changes), but you can't easily split and combine MIDI signals the way you can audio signals in a mixer. Filtering everything through the sequencer wasn't an option, because *Vision* only allows one Thru channel — all incoming MIDI data is re-routed and output only to the instrument specified on that channel.

We were able to overcome this shortage of simultaneously available channels by using the built-in cable-routing and data-filtering features of Mark of the Unicorn's *MIDI Time Piece* interface, which allows the sequencer to address multiple instruments on several MIDI cables. A lot of mid- to high-end MIDI interfaces offer this kind of flexibility, but most people never bother with these functions and simply use the box as a multi-cable interface. Among other capabilities, the MTP lets you route data from any of the eight incoming MIDI ports to the serial port on the host computer and/or to any of the eight outgoing MIDI ports. At the same time, you can "channel-map" (alter the MIDI channel of) all data going in or coming out on any of the ports.

The Casio MIDI horn I played was routed through the MTP to my *K2000* with a channel map on the incoming signals, which converted everything to channel 1 (see Figure 1). Brian's MIDI guitar was routed to the K2000 he was sharing with Claus, also with a channel map on it, set to channel 2. Claus played the *K2000* with local control turned on, meaning that it responded directly to what he played on the keyboard. There was no need for the MIDI notes Claus was playing to go anywhere outside of his instrument, so there was no MIDI Out cable from it to the MIDI Time Piece. Luis's and Mike's *dk10* drum pads were routed to channels 10 and 11 in the *K2000* that they were sharing. The only input device routed into the computer (and therefore the sequencer) was Todd's *VL-1*, so that he could play SampleCell from it.

The MIDI output from the computer — that is, the sequencer's output — was routed to all of the *K2000s*, as well as to the *VL-1*, so that the sequencer could send program changes to everybody. To save on MIDI cables, the *K2000* that was handling the rhythm tracks (which took up four MIDI channels), the interruptus samples, and my sax was connected to the MIDI Thru jack of the *K2000* that Mike and Luis were playing. That meant that we couldn't use any of the same MIDI channels on the two synths, and that all of the channels being used on one machine had to be disabled on the other.

## Life in Hell



"Cream-cheeeeez", *The Return of the Son of Monster Magnet*, **Freak Out**
a different "Cream-cheese"
"Oh my hair's getting good in the back", *Who Needs the Peace Corps*, **We're Only In It for the Money**
"I gave you my high school ring", *Go Cry on Somebody Else's Shoulder*, **Freak Out**
instrumental scream, *Brown Shoes Don't Make It*, **Absolutely Free**
"It Can't Happen Here", **Freak Out**
"Rock and roll stars", *Rudy Want to Buy Yaz a Drink*, **Chunga's Revenge**
guitar intro, *The Torture Never Stops*, **You Can't Do That On Stage Anymore, Vol. 1**
"Da-doot-doot-doot", *Plastic People*, **Absolutely Free**
heavy breathing, **Lumpy Gravy**
heavy breathing, *Call Any Vegetable*, **Absolutely Free**
drum fill, *Montana*, **Overnite Sensation**
"If she were my daughter…I'd", *Brown Shoes Don't Make It*, **Absolutely Free**
"What would you do daddy?", *Brown Shoes Don't Make It*, **Absolutely Free**
whistle and squeak, *Brown Shoes Don't Make It*, **Absolutely Free**

The segment of "*King Frank*" in which we built up to a cacophony was lovingly called Sample Hell. It included instrumental riffs, sung and spoken vocals, and more than a few uncategorizable noises. Sample Hell started with a short and simple sequence that sent MIDI program changes to the Kurzweils. There was a brief pause after the drums stopped, both for dramatic effect and to allow Todd to cue the next sequence, then I nodded my head for everyone to start.

We all hit the downbeat together on notes we had agreed each of us would start with. After that, things fell apart immediately. Everyone could play anything he wanted, and we let it evolve organically — some players would lay back for a few seconds, and then add something in response to what someone else did. Our cue to end the section was a big crescendo led by Mike pounding out "I can't stand it!" (from **Live at the Fillmore East**) on his Kat *dk10* drum pads, faster and faster. I raised a fist in the air to cue the next interruptus segment, at which point Luis triggered a hideous laugh from We're Only in it for the Money, followed by Claus playing, "This must be the end of the world!" from **Lumpy Gravy**.

## An international group

The University of Massachusetts, which has five campuses, is a public institution, supported largely by state funds. Students who live in the state pay very low tuition (relative to a typical private college), while out-of-state residents pay somewhat more. For this reason, the great majority of the 12,000 students on the Lowell campus are from Massachusetts.

The Sound Recording Technology program is a little different, however, and it was with some surpise (and no little pride) that I realized this year none of the students in my advanced seminar were originally from Massachusetts (although two went to high school there). They had come to our program not just because it was a cheap way to get an education, but because it was where they wanted to go to school.

Claus Trelby hails from Denmark, and has also lived in Spain and the state of New Hampshire. Mike Verette grew up in a small town in New Hampshire. Todd Baker comes from a small town in upstate New York, while Brian Calicchia grew up in a suburb of New York City. Luis Silva hails from Ecuador, and hopes to go back there to open a studio some day, although his family moved to Lawrence, Mass., when he was a teenager.

The sixth member of the class, Alex Barshevsky (who didn’t participate in King Frank because his wife had their first child in the middle of the semester and I figured he had enough to deal with), grew up in Russia, and came to Beverly, Mass., a few years ago.

Todd quickly started the sequencer again to begin the "trading instruments" section. For this, the sequencer only played drums and sent out program changes to all of the instruments (and turned off SampleCell by sending a value of zero for Controller #7). But from the audience's perspective, all of the sounds were wrong!

The first aural assault was a dive-bombing guitar. But one look at Brian revealed that he was doing nothing during this section: It was Mike who was triggering the sound with a stick in his right hand and sustaining it with his left elbow on the *dk10* pad. Then Luis began to flail away at his pad, but instead of drums, we heard "*Pop Goes the Weasel*" played on a trumpet patch. A Jethro Tull-style flute riff emerged from Claus's keyboard, followed by a plodding French horn sound, compliments of Todd's *VL-1* keyboard synthesizer. I chimed in on MIDI horn with a bit of the sitar riff from "*Norwegian Wood*." Lastly, Brian entered with a dramatic roll of cymbals and drums coming from his guitar, which had been set to trigger a General MIDI drum kit.

We went around that way twice, each of us playing about eight bars, after which Todd stopped the sequence and immediately cued another interruptus sample — "It's outrageous filth," as intoned by some U.S. senator in the "*Porn Wars*" segment from **Meets the Mothers of Prevention**. He then cued the band and the computer for the "finale" sequence. The sequencer restored our original program changes, and we played the head again one last time. But that wasn't the end. There was one more sample to play from "*Do You Like My New Car*?" off of **Live at the Fillmore East**: "You guys are so professional!" blurted from the P.A. to conclude our performance. The 300-seat auditorium filled with applause, although I'm not sure it was out of appreciation or relief that we were finished.

## Final Thoughts

We didn't set out with "King Frank" to prove a lot of points — we just wanted to do something cool, educational, and fun. What we learned is that performing with computers doesn't have to be an either/or situation when it comes to who's in the driver's seat. With a little clever design, control can be passed from human to machine and back again without compromising your musical goals. Also, it's not necessary to re-invent the wheel, have only the very latest techno-gizmos, or be a hardcore code geek in order to do something really innovative: With the exception of BIAS *Peak* sample-editing software and the Yamaha *VL-1*, all of the products we used in "King Frank" have been available for at least five years. All it takes for a project like this is a little creative thinking, and knowledge of your tools, whatever they are.

**King Frank in concert, 13 May 1997:**
(L-R) Two Kurzweil K2000s in rack, the author and Casio DH-100 MIDI Horn, Brian Calicchia and Casio MIDI Guitar, Apple Macintosh Quadra 650, Todd Baker and Yamaha VL-1, (on floor) Mark of the Unicorn MIDI Time Piece, Claus Trelby and Kurzweil K2000, mixer Bill Carman and Mackie CR-1604, Mike Verette and Luis Silva on KAT dk10s.

---

Thanks to all of my students, wherever they are, and Bill Carman.