

BUILD DAY

It's a gray Saturday morning and the MIT campus is dead. Empty courtyards testify that most students are either sleeping off last night's fun, or still attached to a screen after an all-night computer programming session. Some are up, however, and I'm headed to meet them at Build Day, a semi-regular congregation of musicians and hackers who like to make their own musical instruments. Not much of a builder myself, I'm curious to see how overworked yet hyper-creative MIT grad students use their precious spare time.

I approach the Media Lab, a four-story box banded with three black stripes of windows. Light gray square tiles cover the rest—outlined in black with occasional highlights of faded neon pink, orange, and green. It's a spaceship from planet 1985, dry-docked in the middle of campus.

Up the elevator to a second floor conference room, I find a couple of faces buried in MacBooks, seated at a long grayish table surrounded by windows, whiteboards, and projector screens. Dave Merrill, a Media Lab soon-to-be-PhD, sets out a plastic tub of video game controllers and headphones for people to play around with.

"How'd your show go last night?" Dave asks Rob Morris, by day a Master's student at the Media Lab, by night, singer and guitar player in several bands around Cambridge.

"Oh man, people are crazy on Halloween. It's like there are no rules. It's pretty annoying actually."

One of Rob's signature stage toys is a hacked Nintendo Wii-mote attached to his guitar. He's rigged things so that, with its ability to sense acceleration, the Wii-mote records how he moves his instrument, altering the pitch of the notes he's playing. He shows us a youtube of it in action.

At first, Rob's performance looks like standard fare. He's a talented lead guitarist, throwing down smooth and controlled solos, moving both his body and guitar fluidly, like a veteran performer. Then you notice the whitish, rectangular Wii-mote attached near the base of the neck. Switching it on, his movements become more deliberate. He swings the guitar head up and down in a vertical

plane—a practiced motion that the Wii-mote will recognize. The pitch of Rob’s performance bends with the motion, almost like speeding up or slowing down a record. The crowd’s into it, and Rob’s totally lost in the performance.

Today Rob is planning a new Wii-mote hack. He wants to use the device in conjunction with a handheld microphone to do “voice sculpting” a technique that gives the feel of manipulating your voice with your hands. He’s written software that adds vibrato to his voice when he tilts the Wii-mote in space. It works, sort of, slightly modulating the pitch of Rob’s voice up and down with mathematical precision, but he’s not happy with the effect. Real vibrato is more expressive, due to subtle irregularities in frequency and pitch.

“The challenge is to build something that I’d actually want to use...that doesn’t sound too ‘out-there,’” Rob explains.

Dave chimes in, "ever since I started making musical instruments, I've been surrounded by people who liked avant-garde stuff. But, if you become too inaccessible, you restrict your audience too much."

The five of us in the room nod in agreement about the danger of being too “out-there.” In designing these instruments, any physical gesture—clapping, arm-bending, jumping, pounding buttons, etc.—can be combined with any sort of sound, pre-recorded or synthesized. With such possibilities, the realm of the “just weird” is far larger than that of the aesthetically pleasing.

JOE’S CLASS

About four years ago, the Media Lab’s Joe Paradiso started a class dedicated to the challenges of new musical instrument design. Paradiso is a legend in the electronic music community, known for building a custom modular synthesizer from scratch. In pictures, this synth fills a small room—metal boxes festooned with lights and tangles of patch-cables—like a Borg version of the Flying Spaghetti Monster.

The goal of “Joe’s class” was to invent new musical instruments and computer-music interfaces. The students loved it, but lamented that once the class ended, their creations would probably become side-projects on a forgotten

shelf, fossilizing under sedimentary layers of papers, prototypes, and the odd candy wrapper.

After a semester spent troubleshooting circuits and code, the students wanted time to perfect and actually play their creations. "We had all these students that had been learning this stuff together over the semester, bonding, thinking about musical instruments," Merrill recalled in an interview, "We just wanted the fun to continue." In response, Dave and friend Ben Vigoda decided to organize the first Build Day.

The idea was a hit; people dragged themselves out of bed on a Saturday to get together and hack on music technology. Four years later, the faces change, but people still come to build, to play, to share ideas, to hang out, to make bad jokes, and to enjoy a room full of nerds and gear.

STEP INTO LIQUID

Dave kicks off a round of introductions. He's worked for years in computer music at Stanford and MIT and has built a menagerie of devices along the way. A typical Merrill creation is the Flexi-Gesture—a plexiglass maraca made of circuitry with a handle on top. Sections of the device flex and twist. Bend a section, bend the sound's pitch, twist it and you can rewind or fast-forward—or switch the assignments to whatever effect or sound you want. The player is free to match physical gestures with sound.

As with most of the creations at Build Day, Dave's instruments have both hardware and software components. A common challenge is to get the non-musical tasks, like writing code, to fall in the background, leaving the musician free to focus on performance. This requires intuitive and easy-to-use hardware and software interfaces. To that end, today Dave wants to work on making a new user interface for Pure Data, his favorite open-source musical programming language.

Dave thinks that "PD" is a great programming environment, providing things like oscillators and frequency analysis tools to build software-based musical instruments. But, he points out, it's very hard to "just figure things out as you go",

as is possible with many other music programs. Dave thinks an intuitive interface will make PD more competitive with expensive rivals.

Next up is Josh Rosenstock, a professor of media and game-design at Worcester Polytechnic Institute. He heard about Build Day through Dave's website and became intrigued.

"On the artist-to-engineer spectrum, I'm definitely on the artist side," Josh explains. "A lot of my tech skills are pretty much self-taught, which means I end up getting in over my head on every project I do. So I'm always looking for other people to bounce ideas off of."

Josh has brought with him his "remix jacket," a purple and paisley patchwork coat spotted with microphones and sensors. With the coat on, you can speak into microphones embedded in both the cuff and collar. The sound automatically loops, and you can record more sounds on top of the first to create multi-layered compositions. Once captured, the composition becomes yours to manipulate using your elbows, shoulders, and arm-movements. You might bend the pitch of the sound by bending your elbow, or stutter it by shaking your fist.

Today Josh is working on a similar piece designed for video-remixing. He hopes to incorporate more accurate sensors so that his performance can have more nuance. He just got a new force-sensing resistor in the mail; it measures how hard you press and outputs a proportionate signal that can be used, for example, to smoothly slow down the playback of a video. That's just one possibility of course; Josh doesn't want to figure out exactly how he'll use it until the sensor is hooked up and working.

He also needs to configure his brand new MacBook Pro—which he shows off to the room like a proud father. The ensuing chorus of "ooh"s reveals a strong undercurrent of gear-envy.

Next up, Andres Lombana, a recent graduate of the MIT Comparative Media Studies program. He's been an electronic music producer for nearly a decade, and like almost everyone present, has learned what he's learned by doing. "I don't have a very rich technical background," he explains, "I'm mostly self-taught." Today he's learning to use and control solenoids—electromechanical

pistons—to strike an anvil for an art installation. To the room’s dismay, he hasn’t brought the anvil with him, but does have a bunch of LEDs, alligator clips, and an Arduino microcontroller.

The Arduino seems to be everywhere in the Media Lab. A two-inch by three-inch rectangle of green circuit board, densely packed with chips and solder, it has no particular visual appeal. It’s notable, however, because of how easy it is to use. With a bare-minimum of coding skill and a few sensors, you can have your own novel musical device up and running, connected via USB to your computer, in no time. Andres plans to use the Arduino to control and coordinate his solenoids to create rhythms when they strike the anvil.

Last up is Rob, who admits to not having much physical hacking experience. "I'm pretty bad at hardware," he confesses. He’s got some interesting ideas, however -- like a guitar for cats. "I figure, why should humans get to have all the fun?" he explains as the room busts up at his unveiled prototype.

Rob’s taken a Guitar Hero plastic guitar frame and put six strands of thick rope on the neck, each strand connected to a piezo-resistor that can sense the strain on the rope. It looks like a scratching post attached to a toy-guitar body. The idea is that when the cat pulls on a rope, the action triggers a pre-recorded guitar-lick (performed by Rob). It’s all based on the Arduino board again, but unfortunately Rob fried the circuitry during testing. Not being an electronics expert, he’s hoping to unravel the mystery at Build Day.

"I never really did any hacking before I came here," Rob says.

"Yeah, I used to think electronics were so opaque," says Dave. "I'd look at a circuit board and it seemed incomprehensible." I’m reminded of my college electronics class—my lab partner and I would often resort to randomly connecting different circuit elements while looking at digits on a voltmeter, desperately trying to wrap our heads around the circuit’s design.

Chuckling, Rob continues, "I still don't know what I'm doing. I just plug shit in and if the numbers change in a way that make sense to me, then that's cool."

When the laughter subsides, he continues, "I guess it's the difference between 'hacking' and 'designing stuff,' like for mass production. It doesn't have to work for every temperature and pressure, under all circumstances."

Introductions done, people get to work. Dave's lent me an old gray plastic controller from the original Nintendo Entertainment System console in the 80s. It's been retrofitted to work as a USB video-game controller, but I'd like to make it work with one of my music programs.

By hacker standards, this isn't much of a task. Just convert the signals coming from the controller into a form that my program likes. I open an application that takes game controller input and spits out keyboard commands that my music program allegedly will recognize. It doesn't work of course, so I start poking around in sub-menus and configuration screens to see what's gone wrong.

SHAVING YAKS

As the morning progresses a few more people roll in. Dave's friend and Build Day co-founder Ben Vigoda appears and recruits us all to come down to his car to grab equipment. He's brought three guitars, two amps, and backpacks and tubs full of cables, wires, controllers, and effects pedals.

We tote them up to the conference room and lay everything out. By now the table is covered with all sorts of gear, each person's laptop creating a bay of open space in which to maneuver. It reminds me of when, as kids, my friends and I would dump out all of our Legos on the floor, even though we knew we weren't going to use them all.

Laying out all the options can lead to distraction, however, and Dave recites for us a standard cautionary tale in the Media Lab canon:

"So this guy, an MIT student, sits down to update his resume. He opens Microsoft word, and he starts typing, but he gets about a paragraph in and says, 'you know, I really should be doing this in Open Office instead of Word'. 'Open Office requires Linux, and I have this old PC sitting in the closet that I've been meaning to install

Linux on...' So the guy starts to install Linux on this old PC he's got, but of course the PC needs some new part, like a power supply or something, so the guy starts rummaging around in the spare parts bin and gets distracted by something else...and you can imagine where this is going now. A couple of hours later, his friend calls and asks what he's up to:

'Well...I'm shaving a yak.' "

The joke hits home. It seems too easy to get lost in the details of implementing one small piece of a project. Curiosity wanders and since no one has ever built what you're building now, designs and goals change on the fly. If the point is technological innovation, then there has to be tolerance for wasted time and effort.

Ben sums it up, "Like 90% of our time is spent yak-shaving."

A CONTINUAL STATE OF REVOLUTION

A couple of weeks later, Dave and Joe Paradiso, godfather of Build Day, are talking about the problems of being both the instrument designer and the only person in the world with a clue as to how to play it. Too often, they agree, people spend their time building instruments they never learn to play with virtuosity.

Dave explains, "we always want to have some kind of jam session at the end of each Build Day, but it's always...the thing isn't quite making sound yet...or..."

"Yeah, I know what it's like...it's hard to make an interface and be virtuosic with it right away," Joe interrupts. "The thing is...it has to be stable, and people have to hang with it for a while," he continues. "But you end up taking apart what you have because you want to play with it some more."

Joe goes on, explaining the different mindsets required for building and playing. As the builder, it's hard to "flip the switch" into becoming the player, who must ignore any minor bugs.

"You always want to go back and fix it, which is the easiest thing," he says.

In an influential online article about musical interfaces, Paradiso captured the essential problem. He wrote that advancing technology rapidly displaces old ways of making sound, so "the design of appropriate interfaces is therefore in a continual state of revolution, always driven by new methods of sound generation that enable (and occasionally require) expression and control over new degrees of freedom."

Once, instrument forms stuck around long enough for players to get good at them. Violins, for example, may have changed slightly over the years, but the techniques required to play with skill have changed scarcely at all.

Modern digital instruments seem too fluid for virtuosity to coalesce. On the one hand, they can make any sound, and be designed so that even a complete novice can play tolerably, maybe even skillfully. On the other hand, the technology changes so fast that new options are always available, and the temptation is to incorporate them, which sends the would-be player into a new loop of trouble-shooting and potential yak-shaving.

The problem, of course, is not limited to music, and not particularly new. The constant tinkerer, the obsessive artist, the weekend mechanic—all have their own species of yak to shave. But musical interfaces represent a cutting edge of human-computer interaction, which, as any ATM, automated ticket kiosk, or iPhone will tell you, is becoming more of an issue in our daily lives.

Merrill explained, "If you build an interface that feels good for music, you're already ahead of the curve for building a general, expressive, delightful interface." Merrill uses that word, *delightful*, frequently in describing how humans and machines can best interact. Apple's iPhone is the gold standard. Scrolling the screen with your fingers feels "tight" and the little bounce that happens when you hit the bottom of the page, though almost unnoticed, completes the experience. It has the right combination of responsiveness—it does what you want, when you expect it to—and subtlety—subconscious details, conspicuous only when absent. Dave thinks similar design principles will have increasing importance as computers continue to saturate our lives.

"If we're going to tolerate being surrounded by more and more computers," Merrill concludes, "[they] are going to need to be either less obtrusive, or more delightful."

KOYAANISQATSI

As Build Day wraps up, our ranks have swollen to ten. One guy is working on a hybrid guitar-accordion. Another is working on an iPhone application that allows you to leave virtual graffiti in specific locations using GPS. A third newcomer is working on an effect that turns sound "inside out", reversing high and low frequencies, sort of like taking the negative of a picture.

In walks a guy with an electronics-filled cup attached to string around his neck, turned outwards to give the appearance of a speaker coming out of his chest. He explains that it's a portable audio-looping device built into a paper cup, part of his "Looperman" Halloween costume from the night before.

The room is filled with chatter, the odd guitar pluck, and other semi-musical noise. Someone has put on the movie Koyaanisqatsi, projected on a screen in the background.

The 1982 Gottfried Reggio masterpiece is a non-narrative collage of natural and manmade landscapes. Sped-up clouds drift over mountains like water, brake lights pulse through downtown highways like blood cells—all set to haunting strings and quirky synthesizers of Phillip Glass' soundtrack. Its final scene, an interminable slow motion shot of a piece of rocket, aflame, spiraling back to earth, hammers home a warning against technological hubris.

The Build Day-ers, however, are more focused on their projects than the film. They're making progress, but I can tell we're not going to have a jam session today.

Josh has hooked up his force-sensing resistor to a red clicking switch that clicks faster the harder you push on the sensor.

"Oh yeah, this is Rock n' Roll right here," he grins ironically. It's pretty unimpressive in its current form, but the major problem, getting his circuit to recognize and interpret the signal coming from the sensor, has been solved.

A couple minutes later, Andres announces, "I think I'm pretty done. I just need to build the whole thing now." We all laugh, but know what he means. He's got the Arduino driving an aluminum-encased solenoid with a regular beat of about once every two seconds. The piston drives forward, hits a small clear plastic tub—a proxy for the missing anvil—then retreats slowly before driving forward to strike the tub again. After a day of hooking up the solenoid and wrangling both resistors and computer code to make it work, Andres seems pleased with his progress.

As for my own small challenge, I've figured out how to get the NES pad to work with my program and am playing pre-recorded drum snippets through my headphones using the red "a" and "b" buttons on the controller. It's pretty basic, but I'm stoked that it works. Of course, as soon as it's functional, I start to mess with it, to see what else is possible.

I start to gather my stuff when Jeevan, the new guy working on his iPhone graffiti app, shoots out of his chair, propelled by an unseen force: "It works! I'm a genius, I'm unstoppable."

Dave smiles, "That's what Build Day's all about."