June 2, 2017 - Taeyoon Choi believes the elegance in the abstraction and repetition of computation can only be described as poetry. In 2013, he cofounded the School for Poetic Computation, whose motto is "more poetry, less demo." It's a hybrid of a school, residency and research group that explores the intersections of code, design, hardware, and theory. Choi lives between New York and Seoul, making art projects, teaching future teachers, unlearning the wall of disability, hacking, skating, cooking, and drawing.

As told to Laurel Schwulst, 1854 words.

Tags: Art, Technology, Poetry, Education, Identity, Politics, Process.

Taeyoon Choi on drawing, teaching, disability, and the difference between work and project

Drawing

I draw in order to understand. So when drawing about computers, it's not because I want to illustrate but because I like taking complex things apart. For me, drawing is a way of de-mystifying many of these complex ideas. There's a lot of play and curiosity with drawing. When I hear a talk or when I read a complex book, I always ask, "Would I be able to draw that? Would I be able to understand it if I draw that?" It's not necessarily an end, but it's a process.

I find commercial software kills my creativity. It's designed to do something very professional, like a production tool. In the beginning, I felt like I couldn't really go underneath and truly understand its interface.

When you're learning to paint or sculpt, you learn to understand the material. It's important to see how to make your own paint, or how to carve your stone, or do ceramics, to work with clay. It's really, really physical. I wanted to have that sort of experience with technology, especially the computer. I think I'm most creative when looking at electronic signals and seeing what's happening inside a computer. I'm sculpting with signals. I understand what's happening, and I can control the behavior of my robots or sound installation. It's very immediate. Unlike commercial software, like Photoshop, nobody has decided things for me. It's an open plane to be creative. I can understand every single step. Even though sometimes I don't do every step when I work with other engineers and collaborators, I still have an appreciation for how things work so we can collaboratively be conscious of our decisions.
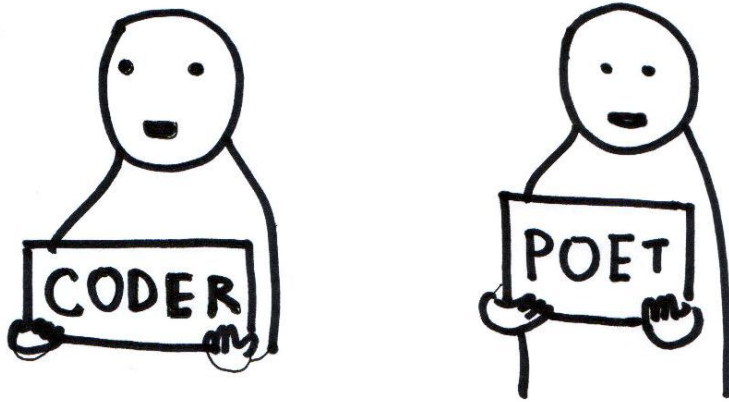
Teaching

The School for Poetic Computation (SFPC) began for a couple reasons:

The official story is that we, the co-founders, were not satisfied in teaching at a big university. We wanted to make something very transparent and very lightweight because the big university is like a black box. That is, students pay a lot of money, but teachers don't get paid very much. Also, knowledge seems to be in the similarly opaque black box between input and output. So we started school from scratch.

My personal story is that I was inspired by Occupy Wall Street and participated in projects like the Public School New York, an open source school. That was a way of looking at how community could be formed through learning. For me, SFPC is a continuation of that activist practice.

The school's motto is "more poetry, less demo." Teaching in any combination of art and technology usually focuses only on making things work or demonstrating a technique. Instead, we wanted to look at the fundamental aesthetics and poetics of code and technology. Poems are not meant to be functional. Poems are more exploratory, and poems are expressive and very human. We looked at code that way.

I think I learn very slowly. I'm not a fast learner. It took me years to learn very basic code. I still feel very naïve when I'm coding. I think most people learn that way, and we as teachers are often forced into situations where we need to be very efficient with our lectures and our assignments. That's really stressful.

SFPC is very unique: it's a 10 week program, so it's intense. People learn a lot in those 10 weeks, but I have a longer term goal. I'd like it to become something like a post graduate program so that students who have done the program can return, preferably for three years or more, where they could truly develop their own practice but also collaborate with me and the other teachers with our projects. There is reciprocity that can only happen in duration. That's often how engineering research works in lab situations. I think art could be done like that, too.

Teaching Teaching

I think the common mistake of a beginner teacher is giving too much: too much preparation and too much energy, too much love. They end up feeling burned out easily.

I try to limit how much I prepare and leave room for students to fill that. I've had good success with that recently. Also, I think teachers are often exhausted even before the class begins simply because of mental anxiety. Before class, I try to sleep well, eat a good meal, and be energetic.

Acknowledging that I'm not an expert always really helped me as well. I don't have answers for many student questions, especially technical questions. Conceptual questions can be confusing too. Sometimes it's good to say something like, "I don't know, I'm sorry. Let's look at it together." This relieves a lot of burden from your shoulders.
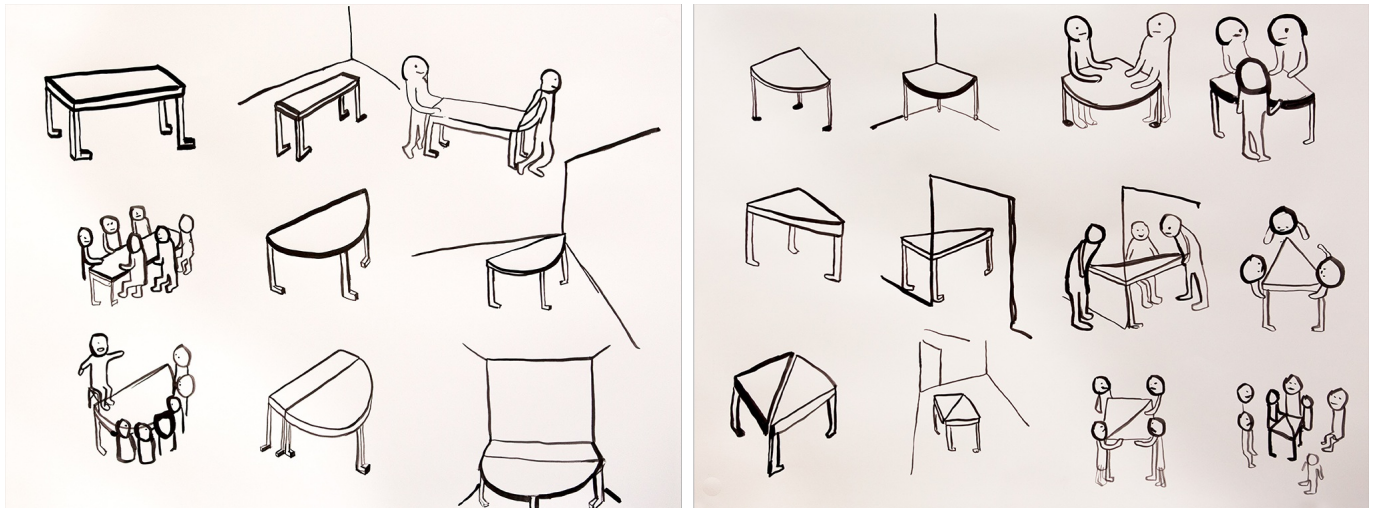
Furniture for All Occasions, 2015.

I learn best when I see a teacher working on a problem. If you have a code problem, you're stuck, and you have an error, what do you do? You open up Stack Overflow. How do you search it? How do you fix things? Learning how to work through a problem is way more important than doing it the "right way." I try to teach the emotional roller coaster of coding, which is similar to the experience of working on art and having breakthrough moments.

There are many different kinds of teaching. It depends on context. For SFPC, we select students so that we can collaborate with them. The focus is finding and building a community we are excited about. When I teach elsewhere, sometimes it's more technical and transactional. I think that's fine; schools have different goals.

Lately I'm trying to focus on supporting future teachers. I think I'm quite good at helping people teach. I taught a class at NYU called "Teaching as Art" and want to do more of that.

I teach because I want to be a student. I still go to classes a lot. Right now I'm learning American Sign Language, which is completely changing everything. I also do yoga. My yoga teacher is an amazing teacher: very generous and supportive. My teacher also takes other teachers' classes. I think reciprocity is about always learning and respecting another. That's the only way you could actually offer something.
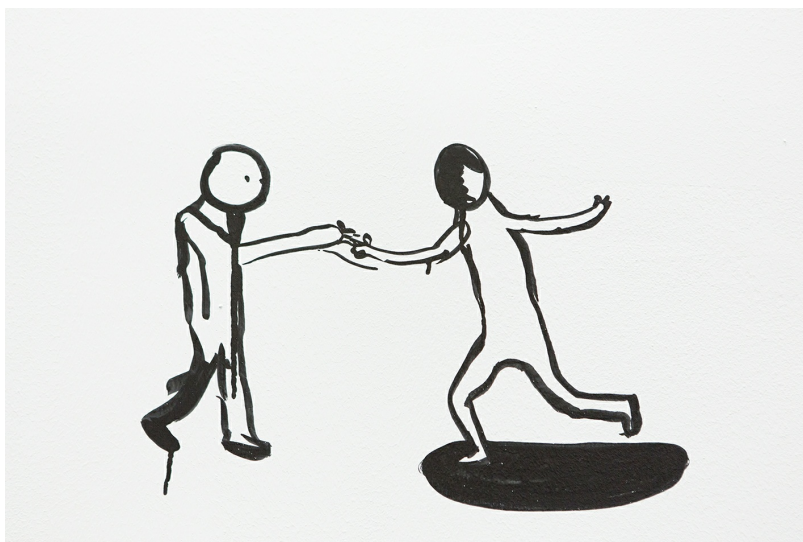
Interdependence, 2016.

Disability

I started another school called the <u>Uncertainty School</u>. It is a collaboration with other artists, people with impairments, and people who work with the disability community.

Although the resources are everywhere online, learning technology and code is culturally not very inviting for people with disability. Deaf students in particular understand language very differently, and the metaphors we use to explain code don't make any sense.

The first iteration of the school was in Seoul last year. We worked with an organization there that supports artists with disability. We had an open call and invited them. Some of the participants have autism, some are deaf, some have a mobility impairments, and some are people with Downs syndrome. However, all of these categories become unimportant quickly. Once you start to work with them, you realize they are people with a slightly different way of communicating or moving.

We tried to translate the coding language, metaphors, and ways of thinking. I designed a bunch of teaching tools, like playing cards and boxes. It was really, really fun. I think it was my most successful and meaningful project.

Disability is a culture. The worst thing is to exploit a community or to appropriate somebody else's culture. So the way I've approached this is that I am a facilitator, more of a support structure for them. I ask them what they want to learn and what kind of projects they want to do, and then we do it together.

Seven Futures, a collaboration with Christine Sun Kim, 2017.

Thinking about disability in the context of SFPC, I realize more and more the opportunities to change how we do admissions and decide to support certain artists or teachers. I realized how flawed and exclusive the traditional academic system is. I'm not saying that SFPC is inclusive or diversified yet, but it's something we are working on. We definitely think about gender, race, and class, but I think there is less visible discrimination against people with disabilities and impairments. It's less visible because it's not taken into account yet, and it's not yet considered in the mainstream.

Work vs. Project

I've been thinking a lot about the distinction between work and project. I read <u>an essay</u> by Paul Chan, *On the Difference between a Work and a Project* in Art Journal 70, No.1, (Spring 2011) which inspired me to think of my own interpretation.

I think "work" is the natural outcome of your everyday practice in whatever medium. It doesn't necessarily mean you need an end goal; you're exploring. You're working on a piece of writing. Work could stand on its own, like a painting or a code. So "project" is where the work becomes consolidated into a vision you have, like "I want to achieve this" with these set of works.

When I work with a certain community on a specific project, I'm not saying that I am that community. I enter that community and do something with them, and then I exit at some point. In the first few years of

my practice, I felt guilty leaving. But now that I know there should be a clear entrance and exit, and that's what defines a project, I feel better about working on many different projects at the same time.

On the other hand, the work, the practice itself, is really intimate and personal. That is very much about what I do and my private space. There's a lot of stuff that I don't share online, work I make alone in the studio.

When you finish a project, you move onto a different project or community or site. We're always entering and exiting. People used to call site-specific artists who would show up and do something and then leave "airport artists." However, as long as it's clear that it's a project, and the artist is not saying they represent that community, the work can exist. I represent myself. I have certain kind of political beliefs that I make clear in my work and practice. That is my work, that is my life, and that's the practice. I will defend that.

Balancing "work" and "project" is important. I have works and I have projects, and sometimes they go back and forth. Life as a practice is important. But projects are not life.

Taeyoon Choi recommends Korean restaurants in ny:

Chodanggol, handmade tofu. www.cdgnyc.com

Hip, fancy, contemporary Korean. www.atoboynyc.com

Woorijip deli supreme. www.woorijipnyc.com

Top class veggie delights. www.hangawirestaurant.com

Korean-Chinese cuisine. www.eatdongchunhongnyc.com

Name
Taeyoon Choi

Vocation
Artist, Educator, Activist, Organizer, Coder

Fact
Taeyoon Choi believes the elegance in the abstraction and repetition of computation can only be described as poetry. In 2013, he cofounded the School for Poetic Computation, whose motto is "more poetry, less demo." It's a hybrid of a school, residency and research group that explores the intersections of code, design, hardware, and theory. Choi lives between New York and Seoul, making art projects, teaching future teachers, unlearning the wall of disability, hacking, skating, cooking, and drawing.

The Creative Independent is ad-free and published by Kickstarter, PBC. See also: Terms, Privacy Policy.

↑