Sara Hendren:

Welcome to episode five of Sketch Model, an audio series about the engineering classroom and how the humanistic disciplines of the arts, the humanities, the social sciences, shape the why and should questions about the technologies we build. Amon Millner is associate professor of computing and innovation at Olin College, my colleague of many years, and he runs the EASE lab there. That's E-A-S-E, which stands for extending access to STEM empowerment. He trained at the University of Southern California, at Georgia Tech and at the MIT Media Lab where he was the co-creator of the Scratch platform, which is a visual and graphical medium for teaching coding widely used in playful ways among young people like my own children.

Amon is dedicated to making the technical prowess in his field, that's human computer interaction, into a much more accessible medium. And I mean literally more accessible. He and his students are known to drive around with a customized trailer behind them, a small coding lab to go, and pull up to a kid's park right behind an ice cream truck in the summertime.

Amon Millner:

And the ice cream truck had a lot of young people around it and they're excited, eating popsicles. And the excitement for eating popsicles is the kind of excitement we want for STEM activities. So we have activities where we take those popsicle sticks and we put them together to make a compass, to make a functional caliper, something that can measure. We slice them and connect them to make a plus shape so that they can be turned into screwdrivers. So these are the tools of engineering, except we're doing it in a fun way. We're engaging and we're using the creativity of young people for thinking about extensions of those designs.

Sara Hendren:

We talked about how he got his own start as a young person, why the performing arts mix so well with code and how he's built a technology lab that reaches both college students and school-aged kids. I hope you'll stay with us.

It's so good to talk to you, Amon. Thanks for being here today. I want to just ask kind of how you got into your field, because I want to help our listeners think through their own experiences trying to do research and teaching that feels like it's holistic and integrated. Where there are the things they care about in the world as a citizen and civic actor that also show up in the very kinds of laboratory outputs and technologies that people make. And so I know people have a lot to learn from how you've structured your career. So can you tell us about how and why you got into programming as a young person?

Amon Millner:

As a young person, I was able to find opportunities that let me see the multiple faces of computer programming, some of the practical. This can be used as a tool to produce a sign or some mechanism to announce that something is happening in a more animated or interesting way. But it could also be a tool that was not outward facing, but something that allowed you to learn about yourself, how you dealt with challenges. Something that only you would be interested in doing. So things that were just for an audience of me.

As well as things that had the ability to change or influence industries or the way that a certain profession did it's day to day jobs. So it had potential to do changes that were at a very large scale with what was a seemingly small set of very useful things. So I got to see the community level coding, the personal level coding and the production level, what it can do for companies and industries all at an early age. So just having the variety of what computer programming could do was very tantalizing.

Sara Hendren:

I guess I didn't realize you were able to see, here is computing in action in the world. How did that stuff reach you at a young age?

Amon Millner:

As a product of Portland Public Schools, I was there at a time when it was evident that the well resourced individuals in the city were finding greener pastures in the suburbs and the investments that were going into the schools where these students were going. And these students were largely white. On leaving the students in the Portland Public School district, a lot of schools were predominantly black and had fewer economic resources. The proliferation of personal computers were finding their ways to laboratories that were only accessible in the suburban schools.

And I was fortunate to be a part of a school district that realized what was to come was that computational literacy was going to be one of the important literacies for the future because the internet was just beginning at those years. And people saw the significance and the importance. But people did have some of the foresight to see the problems that would come with it if we didn't have people that also had computational literacies that have lived experiences along different intersectionalities. People who are raised in communities of color have multiple languages that they speak at home, access to different resources, different approaches to solving problems.

As a result, some schools in my district were granted computer labs that had comparable resources. Apple to EASE and early Macintosh computers were not inexpensive. They were very costly at that time, but we still had access to them. And I was grateful at the power that they wielded to be able to be a part of my education from grammar, in terms of putting different words on the screen and allowing me to work on my language as a elementary school student, to allowing me to play fun games like the Oregon Trail to engage in history. And gaming as well as number munchers and things that would allow me to explore math as well, as basic and logo, some of the languages that allowed me to figure out how do I modify these machines to create a vision that's in my head on a screen that can be enjoyed by other people or just myself.

Sara Hendren:

So you've gone on to make a lot of different kinds of entry points for young people to discover those same things. But when you were in school, there weren't a lot of playful environments, there wasn't a lot of physical computing, there wasn't a lot of visual structuring the way code works on a screen. So what were the resources beyond the what were pricey computers at that time? What were the teaching tools? What did you as a young person sit down and look at on a screen and how did you learn the basics and what lit up your imagination?

Amon Millner:

And you said the magic word when you asked me about what were the basics, because basic computer programming followed a set of rules that was graspable by very young individuals. If you knew how to count by 10, it was a list of things that would happen for the most part. Where you would put a line number and say, 10, this is what should happen. Say what's your name, and ask for an input. Then 20, say hello, name that was input. And then 30 you are such and such. So just make a joke about the person in three lines. But the way that the lines were numbered, you could see that if you count up, you see the first one that's going to be executed is the one that said line number 10. And you don't need to have a line 11. You could have a line 20 to do the next thing, and then the next line that will execute is line 30.

So just the numbering and understanding order, just like a recipe. If you were making brownies, you know that the thing that's at the top, starts of the process and then it goes, you read from top to bottom. And there's a conversation with you and the machine you're using to create something where you could go back and say, oh, there needs to be a line 15 that does this thing that would allow me to do the step between. I forgot that.

So the way that I was introduced to the basic programming language had a lot of room for fun because you can get something to work with basic bones. And then once you got it working you were excited. You had the ability to go back in and add some things in between the lines, if you will.

Sara Hendren:

Yeah.

Amon Millner:

So it didn't feel like I was having to learn computer programming language. I was borrowing some knowledge or just assimilating the knowledge of computer programming into a structure that I had already had. So it didn't feel like a heavy lift as a child.

Sara Hendren:

Amon got his start in childhood as we heard, and in the public schools. And he spent his career making technical tools, but also making opportunities, especially with kids in mind. And I wanted to know what that looks and feels like in practice. The software and hardware itself, but also the moments of exchange. The ways you build places for people to encounter one another in and around the technology.

So Amon, I want to talk about some of the specific projects that you've been part of to do this work of making coding accessible to really wide audiences. And so maybe we'll start with the Scratch platform. Can you tell us what it is, but also then what it does? What it makes possible?

Amon Millner:

Sure. Scratch is a community and a coding platform that allows people to express themselves using code and images and sounds in a way that's very shareable. So that something that comes out of their head, they don't have too many roadblocks to making an interactive version of it that can be seen by people they want to share it with. How we did that was leveraging some of what was explored with previous projects, like making a language for the Lego robotics, the Mindstorm system, that looked like visual Legos that you could snap together. So there was a metaphor of putting Lego pieces together or putting puzzle pieces together.

You can have a graphical language that can guide people on how it was used so that you don't have to have the burden of remembering what structures to put things in when you type them in text. There were a lot of young people that would get tripped up on where exactly to put a semicolon or where a parenthesis goes or what spacing and indenting needed to happen in text based programming languages. So with visual cues, we would give people commands that also had shape. So if you had a block that had an empty space for a diamond shape to go in it, you would know that if you had a diamond shaped block that connecting those two and dropping it into that shape would complete the sentence.

Sara Hendren:

So when you say a graphical interface, Amon, you spoke before about the way code can function like a recipe. And that it can be this kind of fun problem solving thing where you have all these elements that you have to mix together, but then you can go back and you can tweak it and make it do what you want. So why does it matter? What's the gain of translating, like you said, keystrokes like semicolons and stuff into blocks for people?

Amon Millner:

Computer programs are looking for word pairings. What punctuation is used. What's inside of a parenthesis and what's not. So there are so many things to remember because you have to remember what are the words that are available and what will a computer understand? And even if you know those words, you can mistype them and then the computer wouldn't recognize what you're asking it to do. With a graphical language, with Scratch, we try to give people a puzzle based geometry where the words are already there inside of blocks that may look like a rectangle on a screen. So the word move would be a rectangle, and then there'd be a hole for a number to go in.

And if the number, for example, was in an oval shape, you that I could take that number block and I would put it right there in that oval hole and it would snap together, so you know that you didn't misspell the word. You could move 10 by clicking it in there and then that would be something that scratch could execute. But, it gave you visual feedback about what needs to go where so that you didn't put the number in the wrong place like move steps, 10 times. Or 10 is the number of steps that I'd like to move. You could change the order when you think about it in your head or when you write it down in a text based language. But, with scratch exactly where the number goes because there's a hole that corresponds to the shape of the number blocks.

Sara Hendren:

Right. And that's the kind of frustration you're trying to head off. Right? So it sounds like you're saying Scratch is this kind of intuitive way to teach what is a very rule bound structure. But, because you've got it in these strong visual and colorful interlocking pieces, it's like you're teaching without the barrier of, what is this foreign thing I'm supposed to type one by one. But instead, oh, if I snap these pieces together, stuff happens. But, if people are unfamiliar with Scratch, what is it that a kid might build? A 10 year old walks up to the screen open Scratch, what's inherently interesting to them that they might build and do, make happen, by putting those blocks together?

Amon Millner:

When a young person is interested in building a computer program, it's usually not because they wanted to walk up to a blank slate and build something from scratch. So that's not what the name means. The name Scratch came from the DJ culture where a DJ takes two different records, songs that were media created by some other artist. Yet the DJ's job is to mix the sounds from those records together in unusual ways with some changing of the timing. With taking bits and pieces and putting them in different places, sampling, and making new creations out of media that's being mixed together and scratching. What a DJ does is where the name Scratch came from. So that people have typically seen an example project that they found interesting. They've played something. They've had an interaction where someone created an interactive experience where they see one of their friends literally have their face in a computer program and it's saying something with their recorded voice.

And if you press the keyboard, you can change what they say or what they do and what happens to them in their interactions in the virtual space that they've created. Because they have a part of a computer screen, they have access to images, access to sounds, and access to effects where you can change the color of someone. And for example, one of the built in graphics of Scratch is a bowl of cheese puffs. So if a young person took a picture of their face and put it on the screen and also put an object on the screen next to it that looks like the bowl of cheese puffs, and they can ask you a question. They can say, should I eat this or should I not? And if you have the ability to interact, because the Scratch program can tell if you are clicking the mouse button or if you're using the keyboard.

And if you indicate yes, then two different things can happen in the program. One could be that the person's face starts to turn blue and green and say, I'm allergic to that, I got so sick. Or it can be, that was my favorite food. Yum. Thank you. So you have different things that they can program, but these are not video games or things that you would buy on an Nintendo Switch or an Xbox, that are made for the masses. These are things that actually have yourself or your friends in them and only things that they would do as an expression of how they would react to eating cheese puffs. And so giving people the ability to have those inside jokes, those creations that feature themselves and their friends that they wouldn't be able to get from anywhere else, we wanted to make it easy to put the sounds together to allow people to create that interaction.

So there are times when people start with a blank slate, just like a poet could have an empty paper and just come from their head in their heart and create a beautiful poem. Or people can read the literature that's out there and say, oh, tiger, tiger burning bright in the forest of the night. I like that sort of imagery and using an animal. I'm going to be inspired by that. And that's where the starting point of my poetry is going to be. So we offer those two approaches and we really try to make it easier for people to see inspiration from others that are into the many things that Scratch can support. Animation, creating simple games or interactive cards, you name it.

Sara Hendren:

Yeah. I mean it's just so beautiful the way you've... Just all of the metaphors for interacting you've just named are from the arts. I mean from DJ culture, from writing poetry, kind of remixing visual collages and media online. And I think it's just so interesting because you still hear in classrooms and cultures around kids this kind of idea, is your kid a STEM kid? Are they into science and mathematics and therefore engineering? Or are they more of the arts and humanities type? And there is this kind of pervasive type casting that you still hear just in people's casual language. But, everything you're saying about Scratch is that it's taking this deep creative, remix culture and the capacity to the blank page or to do that kind of sampling and mixing together in that kind of artistic impulse. Make a new thing and then share it. And it's all an on ramp for computer programming.

Amon Millner:

And so that ties back to earlier in our conversation when I talked about some of the on ramps that I took to be able to create these languages. I had to have a start and understand the tools of computer programming in order to go on and then be a person that can contribute to the creation of new tools that allow people to have these experiences in hopes that no matter what type of young person is in front of the environment. Whether they consider themselves a musician or a poet, an animator, or already a coder, that they have the ability to learn more, to explore and understand some of the concepts that are underlying even the most technical languages. Because the rules of Scratch are ones where you can have the fun with the coding, but if you want to make a language like Scratch, then you need to understand the underpinnings of computing.

Sara Hendren:

So Amon, among many other projects, I know since the early days of Scratch, you've created tools and platforms to build bridges into communities as an on ramp to that kind of tech enjoyment and tech literacy. And I wonder if you could just talk to us about one of your projects now, which is the EASE lab. E-A-S-E. Can you tell us what it stands for and then what it is that you all are doing?

Amon Millner:

Okay. So when you're on an engineering campus, you typically, if you're a faculty member, teach courses. Some learning takes place in a classroom on campus, but I also have vehicles in which I engage the students in off campus activities. The off campus activities serve multitude of purposes and the vehicles are literal. In this case, we have a trailer that's parked behind the classroom building at Olin College where we can outfit this six foot by 10 foot trailer that a regular SUV can pull behind it. To go wherever and take some of the tools that are typically found in a college campus that may not be in every community center across that reach the young people and marginalized communities and the community centers there.

So when you have a campus that has the resources to teach classes that involve mechanical design, have computers that are powerful enough to render 3D images and then take those things that are rendered and cut them out of machines to create real world products and prototypes, we package some of those tools up in the trailer. And the students in the class can design how the trailer looks and the activities that will go.

But we try to then engage with let's say a community center and a neighborhood within Boston. And one example of a project that we did involved pulling the trailer up to a park behind an ice cream truck. And the ice cream truck had a lot of young people around it and they're excited and they're eating popsicles.

And the excitement for eating popsicles is what the kind of excitement we want for STEM activities. So we have activities where we take those popsicle sticks and we have the machines and the mechanisms and students that are versed in how do you convert popsicle sticks into interesting instruments for engineering. So get them used to the vocabulary of what is a caliper? How can you take three popsicle sticks, cut it down the middle, connect it and make it turn into a functional caliper? Something that can measure tools precisely, measure things.

So these are the tools of engineering, except we're doing it in a fun way where we bring the kind of computer control devices that allow you to modify things like popsicle sticks to put them together and make a compass. To make a caliper. To slice them and connect them to make a plus shape so that they can be turned into screwdrivers. So these are simple things that we have and people don't think about, just the core tools of engineering. But we can have processes where we're using tools that are not normally available in these community centers, but they're available on the college campus. Bring them off campus and have engagements that are based around fun where people are eating popsicles, we're having fun. We're engaging and we're using the creativity of young people for thinking about extensions of those designs. How can you take these basic tools and combine?

So the students that I'm working with are learning from the creativity of the young people and how do they change the scale, connect two tools together, come up with their own types of tools that we didn't think of so that it's not just about the students that I am teaching in the classroom. It's about having extended communities that allow them to see different context. We are in a neighborhood community center. The parents, the elders and the caretakers that are around can also see that the young people there are also interested in these engineering type things and are capable.

And so it is an option on the table so that later on, these students that may not have had access to robotics competitions like a lot of the students at Olin College have had the privilege of being able to have the buy in and the mentors in their neighborhood that can support that. In places where the financial resources to pay those fees or the people that work in the industrial companies don't live in the same neighborhoods. We try to provide similar opportunities or at least let people know that it's not because there isn't capable creativity like that is where all the innovation can come from. These are people that have superpowers. You need to see it. And hopefully that will create a spark that causes students to pursue it.

Sara Hendren:

As he said, Amon's worked on the Scratch platform on another graphical interface for coding called Modkit and on all kinds of partnerships with his lab, including with Boston's South End Technology Center. His students have made expressive playful tools like a digital puppet show, for example. And critical tools like a solar powered charging station at a bench in a Boston neighborhood that serves as infrastructure for unhoused city dwellers.

His work is such an engaging mix of technologies, artifacts together with those encounters and relationships. But the thing that stayed with me the most was talking to Amon about prototyping. About the iterative process of tinkering and testing that goes with any engineering work and the liberty, and also seriousness of that test and see work. Amon told me that prototyping as a practice isn't just something we do for a job. It can also be for life.

Amon Millner:

When I make opportunities for people to get involved in programming, I try to put the programming in a space that it's adjacent to or directly in another context that makes sense to young people where they have fun. And they are already accustomed to applying certain rules that are there or remixing the rules. Whereas close to a context where they already have comfort and confidence and understand some of the structures and understand that the structures can be played with.

Sara Hendren:

I also love that you just start talking about prototyping and it's like one of the most beautiful things about engineers. Just that sense of possibility that you get something up and running and then you get to go back to it and keep making changes and all the ways that suggests that the world is kind of under construction.

Amon Millner:

I agree 100% about the importance of prototyping and being a part of an ever changing world that can be changed by everyone inside of it. Something that I don't take for granted with young people in getting somewhere and then modifying it is that quick wins are important. Especially for young people, because I also remember what is the fastest way to a rough draft that has some semblance of what you were trying to do? And knowing that it wasn't a complete perfect thing that you envisioned, but a prototype gives you an ability to get to a happy point so you feel like you're making progress. Because if you set out to do a complex engineering task and expect perfection at a first pass, that's a recipe for disaster at any age of engineer.

So early on, having things that allow people to take steps where they get some positive reinforcement, like these steps are going somewhere, is an important thing that I recall from my own experiences and try to support young people in getting those firm footings. And those are wins that within the context of a young person and a device, those can be rare wins in life because a young person doesn't always have a lot of autonomy in their own situation.

I was in a school that was mostly free and reduced lunch. And the situations that people endured to get to school, there was already at many points, places where a young person would wish that something was different about what they had to go through, even if it is doing a challenging walk to a bus stop. Given social or given distance, or given that they didn't have the ability to be driven and dropped off. There's a lot that can happen just on the way to school that a young person could wish was better about their situation, but they have little that they can change about it.

So in these contexts of having something in front of you that you can change and get some positive footing, those are things that provide confidence or control in situations where sometimes that is hard to come by for a young person. Especially across different situations. So the ability to have something that's in your head and get it out and see it coming to life, those are things that I think are rare and precious opportunities with all young people in all contexts. And computing is yet another one of the avenues where that can come from.

Sara Hendren:

Amon's work shows me over and over that it's possible to reinvent, distribute and translate engineering what has been a pretty rarefied world of technology into other forms entirely. To create much more every day and accessible engagements with tools and techniques for the stuff in our lives. The stuff we care about for play and for more serious matters.

And as the series comes to a close, we'll return to the big picture. What's all this socio-technical education really for in the end? And how would we recognize it? One of my colleagues, Erhardt Graeff, has some ideas about this. How the engineering classroom might cultivate what's called civic professionalism, and what it means to practice public interest technology.

Erhardt Graeff:

The students did a lot of introspection when thinking about, okay, at what point are they creating greater potential for harm than benefit through building a tool like this? And working with a type of organization that they were working with that really defaulted to handing over data to law enforcement without any safeguards about what that law enforcement would do with the folks that were being identified. And so the students made the very hard decision to refuse to build that web scraper.

Sara Hendren:

That's on episode six of Sketch Model. And I hope you'll join us.

Sketch Model is a production of Olin College of Engineering, a four year undergraduate engineering college outside Boston, Massachusetts. Sketch Model is an ongoing investigation into the substantive engagement between the arts and humanistic disciplines in engineering education. And it's been supported by the Mellon Foundation.

We spent the last four years running programs at our institution bringing more robust arts and humanities to our campus in the form of residencies, some are fellowships for students and collaborations for faculty and staff. You can read all about these programs and ideas on our website, olin.edu/sketchmodel. That's olin.edu/sketchmodel.

Sketch Model team members are Sharon Breitbart, Kristin Casasanto, Jonathan Adler, Deb Chachra and Benjamin Linder. I'm Sara Hendron. Thanks for listening.