



District
Management
Group

INTERVIEW

What Most Schools Don't Teach: Code.org Hadi Partovi's Mission to Bring Computer Science to Schools

Originally published in *District Management Journal*, v.19, Spring 2016

Teaching kids computer science seems completely fundamental, and more importantly, it's just the best career path you could go into.

What Most Schools Don't Teach:

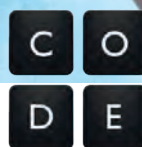
Code.org Hadi Partovi's Mission to Bring Computer Science to Schools

Tech entrepreneur Hadi Partovi, founder of iLike and advisor to a variety of game-changing companies such as Facebook and Dropbox, wanted to make a positive impact on the world and provide greater access to what he calls “the best career path you could go into.” In 2013, Partovi launched Code.org, an education nonprofit dedicated to expanding access to computer science and increasing participation by women and underrepresented students of color.

In his initial video, *What Most Schools Don't Teach*, Partovi enlisted the support of Bill Gates, Mark Zuckerberg, athlete Chris Bosh, celebrity will.i.am, and the founders of Twitter, Dropbox, and Zappos, among others, to encourage more students to learn to code. Partovi was thrust into the education sector as the video reached approximately 20,000 teachers and 10 million viewers.

To build on this momentum, Partovi launched Code.org's Hour of Code, a one-hour introduction to computer science that makes coding fun with exercises that include popular characters from *Star Wars*, *Frozen*, and *Angry Birds*. In less than three years, Code.org has reached over 200 million students through the Hour of Code; even President Obama has participated. Nearly 300,000 teachers have signed up to teach the introductory courses on Code.org's learning platform, and nearly 10 million students have enrolled. Now Code.org is partnering with 100 of the largest school districts to add computer science to the curriculum, and is thereby reaching 10% of all U.S. students.

In this edited interview with DMC CEO John J-H Kim and DMC Associate Christopher Cleveland, Hadi Partovi discusses Code.org's success in creating a groundswell of interest, and how the organization is partnering with school districts to provide students with computer science skills for the 21st century.



Hadi Partovi
Founder, Code.org

How did you first get interested in technology?

My personal story is one of opportunity and the American Dream. I was an immigrant from Iran, and my childhood was spent in Iran during the Iranian Revolution and the war with Iraq. I lived right near the TV station in Tehran, so our neighborhood was literally a bombing target for fighter planes every night. It was not a great place to grow up. However, my life changed tremendously when my dad brought home a Commodore 64 computer that he imported from Italy. My dad said, “This doesn’t have any games on it, but you can learn to code. Here’s a book. Teach yourself and make your own games.” By the time our family came to the United States when I was 12, I’d gotten good enough at computer programming that I managed to get amazing summer jobs as a high school student working at tech companies. I ended up majoring in computer science and having a great career in the tech industry, and I completely attribute this to my early exposure to the field when I was 10 years old.

What gave rise to your interest in promoting computer science education?

Within my own career arc and having been an entrepreneur and a manager at many tech companies, I’ve seen directly how much opportunity is being created in the world of software and how much software is changing every industry, not just the tech industry. I first got interested in this issue because of the vocational angle—seeing the shortage of people accessing

About Code.org

Launched in 2013, Code.org is a nonprofit dedicated to expanding access to computer science and increasing participation by women and underrepresented students of color. Our vision is that every student in every school should have the opportunity to learn computer science. We believe computer science should be part of the core curriculum, alongside other courses such as biology, chemistry, or algebra.

these amazing jobs. I also felt strongly that teaching coding and computer science is a way to address important issues of income disparity and inequality of opportunity.

I saw that the level of learning I got a chance to teach myself is still not integrated in the vast majority of America’s schools. Teaching kids computer science seems completely fundamental, and more importantly, it’s just the best career path you could go into. As a software engineer, you can dream of anything and then build it and then see it in the hands of millions of people and have tremendous impact. It’s such a fulfilling feeling to be able to do that. And we’re not trying to get kids into it because our schools aren’t even teaching it. That seemed to me like a problem we should solve.



STATS FROM
CODE.ORG

Computer science drives innovation throughout the U.S. economy, but it remains marginalized in K-12 education.

- Only **29** states allow students to count computer science courses toward high school graduation.
- There are currently **559,321** unfilled computing jobs nationwide.
- Last year, only **42,969** computer science students graduated into the workforce.



Since its launch in **2013**, Code.org has achieved the following:

232,873,942 have tried the Hour of Code (49% female).

307,182 teachers have signed up to teach introductory courses on Code Studio.

10,243,809 students are enrolled.

25,000 new teachers have been trained to teach computer science across grades K-12.

100 of the largest school districts have partnered with Code.org to add computer science to the curriculum. These districts teach almost **10%** of all U.S. students and **15%** of Hispanic and African American students.

In its online courses, **45%** of students are girls and **45%** are Black or Hispanic. In its high school classrooms, **37%** are girls, and **56%** are African American or Hispanic.

Courses are available in **45+ languages**, and used in **180+ countries**.

What prompted you to start Code.org?

In 2009, I sold my startup iLike to Myspace, and after a short time, I decided to leave Myspace. I didn't want to jump into anything full-time, so I took on numerous advisory consulting gigs: I became an advisor to Facebook, to Dropbox, to another startup that recently went public called Opower, and I joined the boards of several companies. I was biding my time, trying to sort out what I wanted to do next.

If I had to pick the moment that gave rise to Code.org, it actually would be the night of Steve Jobs' death. When Steve Jobs died, every American and, I think, many around the world felt the loss of a great entrepreneur. But, what I saw was the passing of a guy who was only 13 or 14 years older than me, and I started wondering, "What's my impact going to be on the world in 13 or 14 years?"

There's a video that went viral when he died which if you haven't seen it, I highly recommend. It's a version of an ad that Apple never ran, narrated by Steve Jobs, and although I can't quote it exactly, the message is basically, "At Apple, we believe in the crazy ones—the people who are crazy enough to believe they can change the world, and maybe they actually can." And I remember watching that video that night and thinking, "What's my role in changing the world? What if I have only 13 years left? What am I going to do?"

For years I had been brainstorming the idea of making a video to recruit students, teachers, and others to learn to code. I'd always assumed that Steve Jobs would be one of the people in that video. So I started thinking, "You know, I have this idea and I'm not even acting on it, and my potential cast members are dying and I'm sitting here not doing anything." That was the kick in the pants that I really should get going and make the video.

Why do you think that studying coding and computer science hasn't taken hold on its own to the degree needed?

In the mid-1990s when the World Wide Web was exploding with activity, I remember going back to Harvard and asking my dean of Applied Science, "Why isn't computer science part of the

- **9 in 10** parents want their child to study computer science, but only **1 in 4** schools teach computer programming.
- **71%** of all new jobs in STEM are in computing, but only **8%** of STEM graduates are in computer science.
- Women who try AP Computer Science in high school are **ten times** more likely to major in it in college, and Black and Hispanic students are **seven times** more likely.

- A computer science major can earn **40% more** than the average college graduate.
- The Bureau of Labor Statistics predicts that there will be over **1 million** computing job openings by **2024**. These are jobs in every industry and every state.

Harvard core curriculum? Why would we allow somebody to graduate with a liberal arts degree and not have at least a basic understanding of how computers, algorithms, and software work, because lots of other science fields are considered part of the core curriculum. Why not this one?” He actually gave a very interesting answer. He said supply and demand should take care of it. As opportunities increase, more students will go into the field and more schools will be teaching it.

And this was something I kept thinking back on as I was working in the field. It had become clear to me that K-12 education does not react to supply and demand. I’m not even sure if higher education reacts to supply and demand, but K-12 systems definitely do not. They’re driven by state policies, and their goal is to teach foundational stuff to eight-year-olds and ten-year-olds and thirteen-year-olds. The feedback loop of knowing whether we are teaching nine-year-olds the right things so that they can get into the right careers is very slow. And, in general, I think there is this tension that exists in education. Everyone wants kids to have opportunities, but educators like to think that they’re educating kids rather than providing vocational training.

Is that why you named the first video *What Most Schools Don't Teach*?

The plan wasn’t to create an educational organization with a mission to bring computer science to schools. My initial plan was just to make a video to popularize this issue, and then I’d figure out what would come afterward, if anything.

The reason I even got the web domain Code.org is because I’d recruited Mark Zuckerberg, Bill Gates, Jack Dorsey, and will.i.am to be in this video and I figured I couldn’t just put it out on my own personal YouTube channel. With such notable people involved, I needed a website, so I spent some money to buy a great name. But there was no plan for Code.org other than to just put a video on it.

For the video, I wanted a name that would get people to click on it and share it, so I was optimizing for both the title and the thumbnail that shows on Facebook. Three days before the

Teaching kids computer science seems completely fundamental, and more importantly, it’s just the best career path you could go into.

video launched, I ran a bunch of teaser ads on Facebook with a matrix of five different thumbnail images and five different titles to see which combination would get the most clicks. We ran 25 different ads with every combination of titles and thumbnails. “What Most Schools Don’t Teach” with the photos of Mark Zuckerberg and Bill Gates got by far the most clicks, and that’s how we chose it.

For our first video, there wasn’t one Code.org video; there were two Code.org videos. We weren’t sure which would go viral, so there were two different cuts of the exact same interviews. One was five minutes long; the other was nine minutes long. We launched both simultaneously and then we looked to see which one actually got shared more, and we made that the video and the other one became the backup. Code.org has always had a culture of a tech startup, and I regularly say that we’re a tech startup first and an education nonprofit second. We’re a mix of both. We do things differently.

How did you move from that first video to the *Hour of Code*?

What happened is the video got about 10 million views and the website had a petition for people to say, “I think it’s important for students to have the opportunity to learn computer science”—it’s the same petition we have on our site today. If you filled out the petition and you were a teacher, we had the option for you to say you want more information about this for your school. In the first week, we had somewhere between 15,000 and 20,000 teachers reach out saying, “We want this in



our schools.” I had originally thought we would get hundreds or maybe a thousand teachers and that I would connect them to local nonprofits. But when we got 15,000 to 20,000 teachers that first week, we had no idea how to move forward. Code.org was a one-man operation with a two-page website! Those teachers were the inspiration for building today’s Code.org.

I remember thinking, we had engaged 10 million people and tens of thousands of educators to watch this video, and the only follow-up we had was to have them sign a petition. What could I possibly do to get 10 million people to pay attention to this issue again, and what could we do for these 15,000 educators? The answer to that question is how the Hour of Code came into being. The idea was to get 10 million people to do one Hour of Code. I picked the number 10 million because we if we got 10 million to watch a video, maybe we could get 10 million again.

We decided to align the Hour of Code with the next Computer Science Education Week, which gave us a six-month lead-time. We wanted to build up tutorials, develop marketing campaigns, have grassroots outreach, but most importantly develop the follow-through. The follow-through this time wasn’t just to get people to try the Hour of Code, but to

have this be the first step: if you’re a teacher, go on to teach computer science; if you’re a student, take an online course; if you’re a parent, get active as an advocate, and so on.

But should everyone really be learning to code? What’s the rationale?

I realize only maybe 3% or 5% of Americans are going to become software engineers, but the other 95% of people in this country should still have some basic understanding of how computers work, how the Internet works, what’s inside an app, what’s an algorithm. People should know these things for the same reason we teach children things like what chlorophyll is or how electricity works. These kids aren’t going to become botanists and electricians, but we just teach it because it’s foundational to an understanding of the world. And that is actually a more motivating thing for me because it’s a recognition that computer science as a field has reached the maturity and the impact of these other fields and it should be included in a liberal arts education for all students.

You brought a very unique combination of resources to launch this effort. You got *Angry Birds* and Disney to allow you to use their cherished characters for the Hour of Code. Why did you feel that was so important?

I had a personal view, which is just from my own experience, that if it was done the right way, not only could computer science become a course, but it could actually become the most engaging, fun, and relevant course for students. Most of what we learn in school is a curriculum that was designed 200 years ago after the industrial revolution. And looking at my own school experiences, so much of it was memorizing information that I probably won’t ever use again, and regurgitating it on multiple-choice tests.

Meanwhile, computer science lies at the middle of multiple things. It’s analytical like math, but it’s also creative like art class and creative writing class. It’s foundational like algebra

Hadi Partovi’s Biographical Timeline



1984

Immigrated to U.S. from Iran



1990–1994

Harvard University
B.A. and M.S.,
Computer Science



Microsoft

1995–1999

Microsoft
Group Program
Manager



1999–2001

TellMe Networks
Vice President and
founding team member



Microsoft

2002–2005

Microsoft
General Manager

or trigonometry, but it's vocational in that it leads to the best dang jobs in the world. And then lastly, it's culturally relevant, the way sports are. You know, kids love sports because that's what they see on TV, but they also love Instagram and Snapchat. You know what I mean?

I started thinking about how we could get the courses to feel fun. And to make them fun, making them feel like a game was a natural plus. It's hard to make a game that teaches you chemistry or a game that teaches you history, but it's easy to make a game that teaches you to code. In fact, when we started thinking about this, there were already a few games out there that taught this stuff. A very popular tool called Lightbot was used even as early as elementary school by computer scientists. Lightbot is an app where you give code to a little robot to make it walk around. I figured Code.org could do the exact same thing, but using Angry Birds or other popular characters.

PopCap with *Plants versus Zombies* agreed first, and then Rovio with *Angry Birds*. After the success of what we did with *Angry Birds*, Disney came to us. Having Disney come to us and raise the discussion of our using Anna and Elsa from *Frozen* was incredible. It happened to be the most popular movie for kids ever. And then the subsequent year, *Star Wars* and *Minecraft* wanted to do it too.

You use the terms “coding” and “computer science.” Can you talk about the difference?

Actually, at Code.org, we almost never talk about coding. There are only two ways we use the word “coding,” and that is in the name of our organization and in the name of the movement, the Hour of Code. Almost all of our language everywhere else is about computer science. And it's important for us to use both words. When people ask, “How do you compare coding and computer science? What's the difference between them?” the analogy I use is that it's like comparing grammar to English literature.

By far the biggest challenge we've had to overcome ... has been the perception or the assumption that computer science and coding is for geniuses.

We don't just teach kids grammar; we teach them reading and writing and literature. Of course, part of that is learning how to read or write a grammatically correct sentence. But the interesting, creative aspect of writing and reading literature is much broader than that. In computer science, coding one particular language and one particular syntax is easy. Learning how to solve difficult problems using that code, making complicated algorithms, designing data structures and user interface design, understanding how the Internet works, and knowing about cybersecurity are all part of computer science. The field is much deeper and broader than just coding, and it has more relevance to more people.

The reason we use the terms “coding” and “computer science” is because “computer science” as a term is more ill-defined. In fact, when we first started meeting with school administrators and said, “You know, we think you should teach computer science,” the common response was, “Oh yeah, we just bought tablets for all our kids.” And we would think to ourselves, “What do you mean? That doesn't have anything to do with what I just asked you!” Schools often confuse computer science with ed tech or purchasing

iLike

2006–2009

iLike
President/COO

myspace

2009–2010

MySpace
SVP of Technology



1996–Present

Various Startups
Advisor/Investor



2010–Present

Taser International
Board Director



2013–Present

Code.org
Founder, CEO

We have always been dealing with schools that want to make this work, so for us, it has always been more of a question of how we can solve their problems.

hardware. Whereas when you ask them about coding, they'll say, "Oh, we don't do that." Coding is the easier, more popularly understood entrée into the field of computer science.

But what we are actually teaching is not just coding. If you look at the actual courses that kids are taking in our programs, we're teaching concepts like loops and functions and variables without even teaching any syntax, and we have exercises that teach how the Internet works, the importance of cybersecurity, how data encryption works, how compression works, how information is stored in binary—all those types of things have nothing to do with coding; they're more about learning how the world works.

What has been the hardest challenge your organization has had to overcome?

By far the biggest challenge we've had to overcome among school teachers, educators, parents, and students has been the perception or the assumption that computer science and coding is for geniuses. We want to get everybody, but especially teachers, to recognize that the students in their classrooms can learn the ABCs and 123s of computer science. Our goal is to convince you that computer science isn't an honors class, it's a kindergarten class. My dad has a Ph.D. in theoretical physics and I remember as a student wondering, "Should I become a physicist like my father?" I learned enough physics to realize that understanding special relativity or general relativity or quantum mechanics is really hard, and computer science is actually much easier. And the basics of coding are way easier—every field starts with the ABCs and 123s.

It's not hard to convince people that technology is changing the world and that it's important. Everybody gets that. But the idea that your nine-year-old kid in your low-income school can actually learn what it takes to get started down the path of being the next Mark Zuckerberg, that's the hard part. The stereotype of someone interested in computer science is a teenage white boy in a dark basement with an energy drink and *Star Wars* posters and things like that. They don't think of a Latina girl who's nine years old in a Los Angeles urban elementary school. Changing that stereotype has been our most important objective.

And some basic logistics of adding computer science to the curriculum have been challenging as well, right?

Absolutely. Things like who's going to teach the course and where will it fit in the schedule are big obstacles when working in schools. Everyone's busy. But one good thing is we have always been dealing with schools that want to make this work, so for us, it has always been more of a question of how we can solve their problems. You know, most ed reform efforts are trying to do things for schools that the schools don't want. Whether it's longer school hours, smaller classrooms, teacher accountability, more assessments, fewer assessments, there are all these different arguments about how to make schools better, but most of those initiatives aren't led by the teachers themselves wanting it. Most of them are led by some philanthropist or government official or whoever trying to impose a better way to do it, whereas we from day one had 15,000 to 20,000 teachers who wanted this; that desire more than anything else has helped us overcome obstacles.

We've also used funding to overcome obstacles, so we would pay for teacher training, and initially, we'd even pay for salaries of teachers while they were going through the training so that every school who wanted to work with us could get a trained teacher to teach the course. We did what we could to make it as easy as possible for a school that wanted to make it work.

So, what's the ultimate goal for Code.org?

The goal for Code.org isn't for Code.org to be taught in every school. It's for computer science to be taught in every school. There is no biology.org or math.org that's trying to get these subjects taught in every school; it's just part of the system. We ask ourselves, "What are all the pieces that need to change for

computer science to become like math and biology in the school system?" We think of ourselves as a change agent, and the end game is for computer science to be taught in every school. This means they have curriculum, teachers, a course schedule that includes it, graduation requirements that encourage it, state policies that fund it, and schools of education training new teachers to teach it. Getting to that end game may take a lot of time, but I also think the momentum behind this is not only big, but is accelerating.

Recently the president of the United States announced that computer science is no longer an optional skill. He says it's a basic skill alongside the three Rs. The president says this not just because he thinks so, but because polls show that the majority of Americans believe computer science is that essential at this point. It shows that we've moved perception enough that people really believe in the importance of computer science, and the momentum behind this is significant.

Now, whether Congress funds the president's Computer Science for All initiative or whether the state governors fund the plan, I think we're entering a new chapter for Code.org because computer science has reached a level of importance that the highest officers are thinking of it as worth pursuing. We as an organization don't think that we can just give up and let the government do it all. I don't think any American looks at our government that way. But we do think government funding will really change the game in terms of the pace. I think there are a lot of schools that want to adopt computer science and don't have the wherewithal to do so. We can help, but we'd like to see the government play a real role in funding that change.

I'm personally hopeful that within five years we'll see our goal realized of every single school in this country teaching computer science. If you had asked me that earlier, I would have said it's impossible, but it's quite likely now.

Are you worried that if Congress gets involved, this will become one of those reform movements that is being done "to you" instead of "by you"?

I'm not worried about that for two reasons. First, because the grassroots demand is there, and second, because Congress isn't looking at it as a mandate. The way it's being pitched is as funds for schools that want to apply for them.



I almost would say it's imperative that Congress pass something just because the intense desire is there in a fully bipartisan way among American citizens, among the governors, among the schools. I'm not sure what will happen, but we're going to stay hopeful for some change.

Do you have a message for superintendents and district leaders?

More than anything else, I would ask superintendents to think of computer science the same way they think about any other academic field that students start in elementary school and learn through middle and high school. This is something that your teachers *can* teach to your students.

The largest districts in this country have already embraced this belief, and many of them have made computer science foundational. In Broward County, the sixth-largest school district, every single high school teaches computer science today. In Charles County in Maryland, every high school, middle school, and elementary school already teaches computer science. These districts started down this path less than three years ago. It's not hard to do; it just takes a little bit of will. You'd probably be surprised to find that in surveys, over 50% of all teachers believe that this should be taught and that it should be required for students to learn. So, your teachers want this to be something that's taught in schools. I hope that administrators recognize this demand. ♦