Do you remember when you wrote your first computer program? I was in the eighth grade, at the Radio Shack in the mall. At the back of the store, TRS-80 computers, as well as books on BASIC programming, had been made available for customers to tinker with. And I was hooked. My parents didn’t understand my obsession with computers at the time, and I must admit that today, as the parent of a teenager, I often find it hard to accept the amount of time my son spends on his smartphone using social media. The difference is that, in the 80s, if you wanted to do something fun on a computer, you had to write code. Today, smartphone apps and video games have perfected the art of entertainment to the point that all free technology time is spent having fun—you can’t program on a smartphone, Xbox, or PlayStation. In fact, most teenagers have no idea what a commend prompt is or what “IDE” stands for. But the smartphone generation is poised to become expert at communicating and sharing with others.

Minecraft, which became a huge hit 10 years ago, was bought by Microsoft for $2.5 billion in 2014. The game’s developers recognized that users would have just as much fun building their worlds as they did interacting with others. Minecraft offers an in-game programming experience through Redstone that allows using visual components for player interaction. Minecraft also has a Lua programming language plug-in to interact with game elements via an in-game command prompt. For the typical Minecraft player, game building is why you play; very little time is spent on programming. At TechGarage, the large

**Scooter Willis, TechGarage**

Students are ready to embrace the technology-driven 21st century. If we make learning fun and challenging—and then get out of the way—they will readily learn the skills needed to succeed in the future.
after-school robotics program I run, the quickest way to find out whether fourth graders are going to love building robots is by asking whether they play Minecraft and seeing if they smile. Minecraft will be responsible for an increased number of architects and civil and mechanical engineers, because building a virtual city in the fourth grade puts a young mind on the path toward a love of building.

TechGarage offers a comprehensive four-week summer camp program in which we’re constantly looking for ways to make learning core engineering skills fun. By helping students find their technology passion at a young enough age, we hope to make a positive impact on their schooling by giving learning a purpose.

Five years ago, we introduced RoboCode, a Java-based tank game developed by IBM in 2001 to teach programming (Figure 1). The game consists of a Java-based tank API that controls all elements of the tank and a collection of tanks with a range of features. The tanks—with names such as Crazy, Corner, Fire, and Walls—come complete with source code, which facilitates student learning because they have to study the code for each robot. Each week, a new group of 15 students was tasked with writing code for their tank, which would compete against the tanks in the game and, more importantly, the tanks of other students in the class. There was no instruction in Java—we simply directed students to the “video game” and let them figure out how to program. Over the years, we introduced hundreds of students to Java by using RoboCode, and were consistently amazed at how quickly they learned to program when faced with the knowledge that they had a competition at 3:00 against their peers. The final double-elimination tournament between student tanks often resembled a Superbowl-watching party, with everyone cheering for their favorite team. If you make learning fun, all you need to do is get out of the way and let the fun begin.

Two years ago, we started using CodeCombat.com, a multilevel video game that doesn’t require a keyboard and mouse to compete; instead, players are required to write code in either Python or JavaScript to navigate their characters through increasingly complex game levels. These novice programmers are challenged to install and write their first program without a system administrator. CodeCombat.com is web-browser-based, and the only difficulty is when parents try to set up accounts for their sons or daughters. The solution is to have parents let their sons or daughters sign up for the account. CodeCombat.com offers six levels of computer science, game development, and website design, and is recognized by the College.

Figure 1. RoboCode, a Java-based tank game developed by IBM in 2001 to teach programming (https://sourceforge.net/projects/robocode).
Board as an endorsed provider of curriculum and professional development for AP Computer Science Principles (AP CSP). It epitomizes the idea of making learning fun and getting out of the way.

Technology is causing evolutionary change in all aspects of society. Discovering optimal ways to educate and prepare our children for a technology-driven future is paramount. Of potential concern is how to enable students from varying socioeconomic backgrounds to participate and work in an increasingly technology-driven world. At TechGarage, we’ve been working on at-risk engagement models to bridge the digital divide by partnering with community centers that support after-school and summer camp programs in at-risk neighborhoods.

At the end of the 2018 school year, TechGarage, together with its community center partners, hosted three 2-week robotics summer camp programs to expose elementary and middle school students to challenge-based learning and robotics. Before and after these camps, a STEM assessment survey was given to 40 students with no prior participation in the TechGarage program.

In the informal survey results from the Boys and Girls Club of Riviera Beach and the Delray Achievement Center, where the self-selected students were new to the TechGarage program, 90 percent of students reported that they liked figuring out why something didn’t work and fixing it—which are core attributes for engineers. Eighty-eight percent said that they like electronics, which was most likely driven by their exposure to smartphones at a young age. It’s our assessment that a smartphone represents a small learning box that presents problems to be solved on a daily basis. In the survey at the start of camp, 30 percent of the participants said they were interested in what makes machines work, and 50 percent responded that they were sometimes interested in this. After two weeks of robotics camp, 70 percent responded that they were interested in what makes machines work; this significant increase drew from the “sometimes interested” group, which dropped to 15 percent, and the “not interested” group, which dropped from 20 to 12 percent. A similar response was seen for soldering circuit boards: wherein, in the initial survey, 30 percent said that they liked soldering circuit boards, this number jumped to 62 percent after the students were given electronic circuit board project kits to solder/make during the two-week robotics camp. Clearly, the additional exposure to technology using an engaging, challenge-based approach generated significant interest in learning more.

For survey questions that focused on interest in programming, we did not expect strong initial interest. Thirty percent of students responded “yes” when asked whether they wanted to program and develop video games. Another 30 percent responded “yes” when asked whether they wanted to become robotics engineers; in the post-survey, this increased to 45 percent. When initially asked whether they wanted a job that makes lots of money, 95 percent responded “yes,” and 5 percent “maybe.” Students were also asked whether they liked programming in Python; 30 percent responded “yes,” despite the fact that informal questioning of the groups found no prior experience programming in Python. More importantly, “yes” responses jumped to 62 percent after two weeks of exposure to CodeCombat.com (which teaches Python in a video game framework), with the change coming from a drop—from 42 to 14 percent—in those initially responding “no.” Similarly, in the pre-survey, 62 percent responded affirmatively to the question “I like programming a robot”; this group increased to 78 percent in the post-survey. These results largely support the premise that students have a strong interest in technology and that exposure to technology learning models increases their enthusiasm and interest.

Teacher-centric methods that emphasize rote learning and memorization run the risk of marginalizing students who don’t perform well in traditional schools, setting them on the path to a minimum-wage future. With the recent passage of the Carl D. Perkins Career and Technical Education Improvement Act of 2018, which received near-unanimous support from Congress and $1.8 billion in funding, we must take a hard look at the optimal ways to prepare students for technology careers.

At TechGarage, we’re convinced that the smartphone generation knows no socioeconomic boundaries—the playing field has been effectively leveled. Students are hooked on technology: they are conditioned to fail early and often from playing video games, learn quickly by watching YouTube videos, and have very little interest in memorizing their way to straight As. Challenge-based learning in the context of technology education presents real-world challenges through collaborative and hands-on programs that develop much-needed skills. Students are ready to embrace the technology-driven 21st century, whereas many schools are still using teaching models developed for 20th-century requirements. If we can make learning fun and challenging—and then get out of the way—students will readily embrace their future.

Scooter Willis is a founder of TechGarage, a centralized robotics workspace for middle- and high-school students in Palm Beach County, Florida. Willis is also the senior director of R&D Bioinformatics at F1 Oncology. Contact him at willishf@gmail.com.