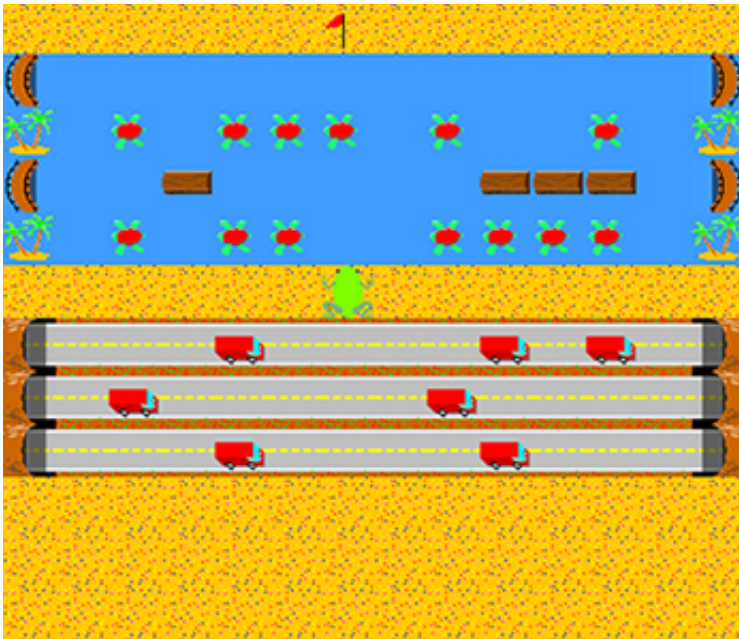# Team uses video games to spark kids' interest in coding

December 12 2013

A 2-D Frogger game created using AgentCubes.

(Phys.org) —In grade school classrooms across the country, students have been hard at work this semester trying to figure out how to smash a virtual frog with a virtual truck.

They're building their own video games—inspired by the 1980s classic Frogger—and there are a thousand details to work out: Should the truck

kill the frog or just knock it unconscious? How may trucks should there be and how fast should they drive? How do you make the frog move and how many lives should it have? What happens when the frog finally makes it across the road without getting hit?

In the end, the [students](#) will have built a video game. But more important, the students will have learned how to code—whether they knew it at the time or not.

The strategy, hook kids with video games and then slip them some computer [science](#) skills on the sly, is the product of two decades of research by University of Colorado Boulder computer science Professor Alexander Repenning, whose goal is to attack computer science's image problem and get more kids, especially girls and minorities, interested in going into the field.

"Programming should be easy and exciting," Repenning said. "But that's not where we are. The perception of students is that it's hard and boring. Our goal is to expose a much larger as well as broader audience to programming by reinventing computer science education in public schools."

For the first time, the video game-building tools Repenning and his team have been using with school kids are now available online. Anyone with an hour to spare and a curiosity about code can build their own 3-D video game using CU-Boulder-created AgentCubes online at [hourofcode.com/ac](#). The tutorial is part of the global Hour of Code initiative happening now during Computer Science Education Week. The idea? Just try it; you might actually like it.

The threshold of understanding

The stakes for increasing students' interest in computer science are high.

By 2020, there will be 1 million more computer science jobs available than students who are studying computer science, according to the computer science education nonprofit Code.org. And even for those who don't plan to major in computer science, coding is becoming an increasingly important skill across a wide range of disciplines.

Repenning sees the problem of interest as an issue of relevance. Coding skills should be taught when students need them to build something they want to create, not just for the sake of learning to program.

Repenning and his collaborators, including Kris Gutierrez and David Webb, both professors in CU-Boulder's School of Education, have built an entire curriculum, called Scalable Game Design, that's based on this principle of just-in-time skills. First, they get the kids excited about making their own game—a task that's not terribly difficult, as it turns out—and then they dole out the computer science lessons when they're needed.

"The notion is that people learn best if you push them to the threshold of their own understanding," Repenning said. "You want to make something, but you can't quite make it because you're missing at least one important piece. You're at that threshold. If I were to give you one more piece of information, you could do it. That's exactly where we want to have them."

Some of the missing pieces are technical. But others are a matter of learning a different way to think about solving problems.

"It's not just programming; it's thinking about the problem and developing a meaningful solution," said Ian Her Many Horses, a doctoral student in computer [science education](#) working with Repenning and Gutierrez. "Computer science is just solving problems, and everyone has problems."

Expanding across the nation

Repenning first rolled out Scalable Game Design at local schools, but the program was a voluntary after-school activity and it attracted the usual suspects: boys. In subsequent years, the project was introduced into computer literacy classes that were already being taught during the school day, exposing all kinds of kids who might not normally be inclined to try computer programming, especially girls and minority students, to code.

"We asked them after, 'Did you enjoy the activity?' And they said, 'Yeah. We love it and we want to do more of it,' " Repenning said.

The program is now ubiquitous in Boulder-area middle schools, and beginning about five years ago, Repenning received a $1.5 million grant from NSF to expand the program to schools outside the local district, especially districts with widely varying demographics, from inner-city schools to extremely rural schools and Native American communities. To implement the expansion, CU-Boulder hosted trainings on campus each summer to prepare teachers to deliver the program.
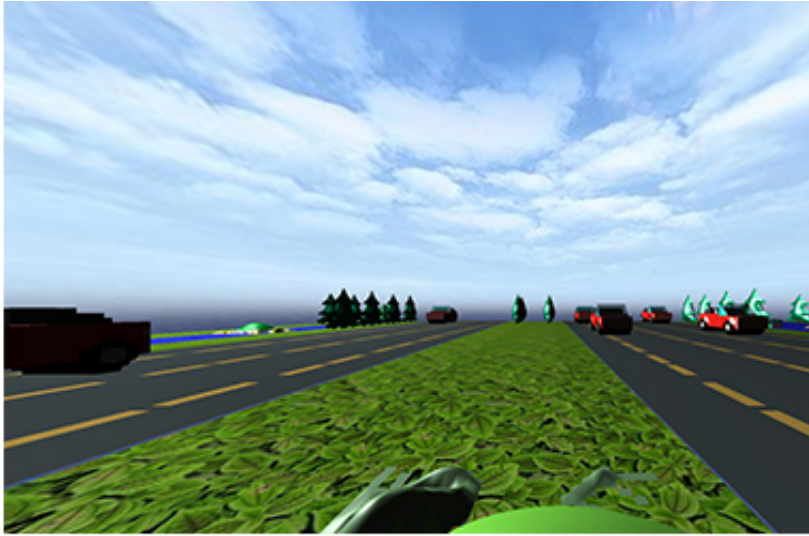
A 3-D Frogger game created with AgentCubes from the perspective of the frog.

During the first expansion, Repenning and his colleagues also discovered a theory of broadening participation suggesting that the just-in-time teaching approach was especially important to getting girls, who are vastly underrepresented in computer science, excited about programming their own games. By contrast, direct instruction seemed to turn girls off. (Boys, on the other hand, remained generally excited regardless of teaching technique.)

Repenning has since received two more NSF grants. The first, for $1.5 million, is being used to follow up on how pedagogy affects girls studying computer science. The second and most recent grant—$2 million awarded in August—recognizes the achievements of the initial expansion effort and is being used to further spread Scalable Game Design across the country by offering teacher training online through collaboration with the eMINTs National Center at the University of Missouri.

Computer science is for everyone

Once students have begun to master the drag-and-drop programming environment created for use with the Scalable Game Design curriculum to actually build a game, the sky's the limit. For example, the same type of code that dictates how a truck collides with a frog can be used to dictate other types of collisions—including two molecules combining in a chemical reaction—which allows students to build science simulations as easily as video games.

And the ability to design all the characters used in the video game as well as the land in which the characters live allows the students to make an infinite variation of games well beyond the Frogger example.

"Computer science is like real-world magic," Her Many Horses said. "Anything you can think of, you can make happen in this virtual environment."

At Sanchez Elementary School in Lafayette, Colo., students as young as second-graders use Scalable Game Design in an after school program run by CU-Boulder called El Pueblo Magico. This semester, fifth-grader Isabel Esquibel and her partner are building a game about a student who is trying to safely travel from school to his home. On the way, the student has to dodge cars, duck wrecking balls and avoid trucks to make it to his front door—and to victory.
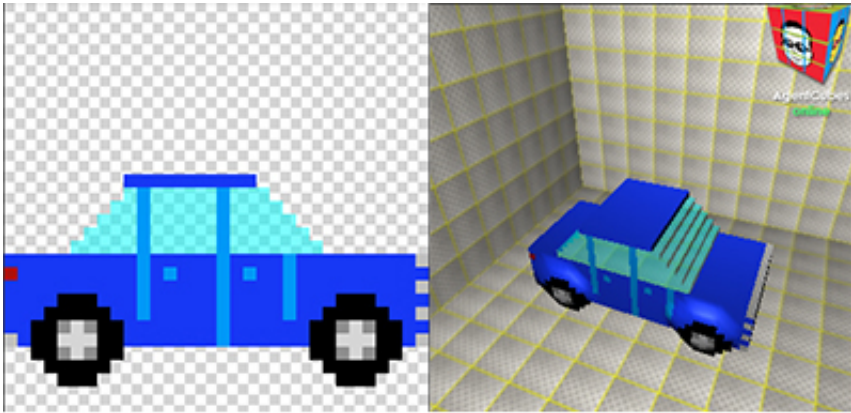
Participants can create 3-D "agents" by inflating 2-D icons.

"It's fun because you're making your own [video game](#)," said Isabel, who now wants to try more coding in the future. "It's kind of easy, but at some points, it gets harder before going back to easy again."

Isabel and her classmates at Sanchez are just the kind of students that Repenning and his colleagues hope will stay interested in computers and bring a greater diversity to the field. At Sanchez Elementary, 67 percent of the students are Hispanic and 65 percent qualify for free lunch.

Her Many Horses, who works with the kids in the Sanchez Elementary after school program, says exposing these kids to coding will open doors for them. "It offers so many more options for the future that they might not think are available to them," he said. "They may think that computer science is just for those students over there because they like math and science, but [computer science](#) is for everyone."

Provided by University of Colorado at Boulder