EDUCATIONAL PROGRAMMING ENVIRONMENTS have tried to broaden the participation of women and minorities in computer science education by making programming more exciting and accessible. Starting in 1991, our research at the University of Colorado with AgentSheets explored the idea of supporting students with game design and simulation building through drag-and-drop interfaces. Early on, AgentSheets enabled students to build and share their creations through the Web and more recently AgentCubes is exploring the idea of 3D fluency through gentle slope 3D. Others followed with slightly different aims. For instance, Alice explored the idea of storytelling and Scratch the idea of animation. However, the point of this column is not to compare programming environments but to explore why programming is still not used in public schools, and particularly middle schools, in spite of these tools.

While the broadening participation situation at the high school level still looks bleak it is much worse at the middle school level. To the degree that programming is found at middle schools at all, it is usually offered as after-school programs. Middle school is an essential period of life during which students, especially girls and minority students, make decisive but often-inevitable career decisions such as “science is not for me.” How can we shift middle school computer science education from isolated after-school efforts to a systemic model in which computer science is integrated into the school curriculum and taught in required classes at districtwide levels? Our Scalable Game Design Initiative, with over 8,000 participants, is showing great initial promise and is formulating a new approach in various settings that include inner city, remote rural, and Native American communities. But before I describe our strategies and results, I will begin with a story.

Six years ago one of our teachers decided to use AgentSheets to introduce game design as a programming activity in his regular computer education class. Previously, his class was limited to topics such as keyboarding and PowerPoint. When I visited his class for the first time, I was truly surprised. I immediately noticed a completely different participant composition. Instead of the typical single girl found in the computer club, about 50% of the students were female—there was also a large percentage of minority students. I asked students if they liked the game design activity. They did. However, many also indicated they would never have gone to a Friday afternoon computer club to do programming. One student summarized her perception of computer science as “hard and boring.” The basically universal excitement about game design in a required class suggested a strategy to simultaneously increase the exposure of student interest in computer science and broaden participation.

Since then, the NSF Innovative Technology Experiences for Students and Teachers (ITEST) program enabled us to expand this experience into a complete strategy for middle school computer science education. We were able to study the systemic integration of a game design-based computational thinking (CT) curriculum into middle schools. The resulting Scalable Game Design framework (see the accompanying figure) includes a curriculum of increasingly advanced game design activities that range from basic classic 1980s arcade games such as Frogger (bottom-left in the figure) to more contemporary games such as The Sims. The framework includes sophisticated artificial intelligence concepts (top-left in the figure). The right-hand side shows our Zones of Proximal Flow dia-
gram (a combination of Csíkszentmihályi’s Flow diagram with Vygotsky’s Zone of Proximal Development conceptualization). The essence of Scalable Game Design is that programming challenges and skills should be balanced and there are different paths, some better suited than others for broadening participation, along which students can advance their skills and tackle more advanced challenges.

The four main goals and approaches of the Scalable Game Design framework are described here.

**Exposure:** Develop a highly adoptable middle school CT curriculum integrated into existing computer education and STEM courses so that a potentially very large and diverse group of children is exposed to CT concepts. The small number of middle schools that offer programming-related after-school programs attract only a small number of students. A successful computer club at a middle school may attract about 20 students consisting almost exclusively of boys already interested in programming. In the Boulder Valley School District in Colorado, a technologically progressive and affluent district, only one of 12 middle schools offers a computer club. With Scalable Game Design introduced to the curriculum, 10 of the 12 middle schools now offer CT classes. Additionally, most of the students participate, resulting in an extremely high participation of girls and minority students. Out of the over 8,000 study participants, 45% are girls and 48% are minority students. In some of the larger schools we have 400 students per year per school participating. It is clear that a curriculum-integrated approach has a much higher potential for systemic impact compared to an after-school program. However, to reach this kind of exposure school districts must see direct value in CT education and find ways to integrate it into existing courses.

The left side of the figure shows some AgentSheets/AgentCubes STEM simulations and games employed in existing courses to address STEM and educational technology standards. STEM simulations require computational thinking tools such as AgentSheets that can run advanced simulations with thousands of objects, include scientific visualization, and provide means to export data to Excel and other tools. We have found that most schools have the capacity to incorporate these activities into existing courses. With over 16 million middle school students in the U.S., the potential exposure of students to CT through this strategy is enormous.

**Motivation:** Create a scalable set of game design activities ranging from low-threshold to high-ceiling activities
so that students with no programming background can produce complete and exciting games in a short amount of time while still moving on a gradual trajectory to the creation of highly sophisticated games. Scalable Game Design has a uniquely low threshold for teachers and students making game design accessible across gender and ethnicity.

We have developed a professional development program based on approximately 35 contact hours in which we train teachers to have students build their first playable game from scratch in about a week (for example, 5 lessons x 45 minutes). The ability to create a playable game is essential if students are to reach a profound, personally changing “Wow, I can do this” realization.

On the right side of the figure, zones of motivation are delineated based on challenge versus skills. These zones are Anxiety, Zone of Proximal Development, Flow, and Boredom. In middle schools we have found the path called Project-First is significantly more effective than paths relying on learning many principles first without the presence of a concrete and interesting challenge. The Project-First path maneuvers students into the Zone of Proximal Development where, with proper support, they quickly learn relevant CT concepts. Motivational levels, as measured by the expressed interest to continue with similar classes, are extremely high: 74% for boys and 64% for girls; and 71% for white and 69% for minority students. In most schools these are not self-selected students. While there are no well-established baselines, our teachers considered a desire by one-third of their students to continue a success.

**Education:** Build computational instruments that analyze student-produced projects for CT skills so that learning outcomes can be objectively measured. These outcomes include learning trajectories and transfer of CT concepts from game design to simulation building. What do students really learn? While CT definitions are still somewhat forthcoming, one school director created a succinct statement of expectation—“I would want to walk up to a student participating in game design and ask: Now that you can make space invaders, can you also make a science simulation?” This question of transfer should be at the core of computational thinking education. If students learn to build a game but have no notion of how to transfer their skills into science simulations, then game design has no educational justification for being in a curriculum. We devised the Computational Thinking Pattern Analysis as an analytical means of extracting evidence of CT skill directly from games and simulations built by students. Most excitingly, this lets us go beyond motivational research and study the educational value of game design, including the exploration of learning trajectories and transfer from game design to STEM simulation building. To enable this transfer, the professional development of teachers stresses these computational thinking patterns as a means of promoting CT.

**Pedagogy:** Systematically investigate the interaction of pedagogical approaches and motivational levels so that teachers can broaden participation. With school sites in Alaska, California, Georgia, Ohio, South Dakota, Texas, and Wyoming and over 10,000 student-created games and simulations, we were able to explore a uniquely rich set of motivational and educational data. We found the main common factor supporting motivational levels and skills across different school contexts, gender, and ethnicity was scaffolding. Of all the factors we considered, scaffolding was the only significant one. Scaffolding, a pedagogical aspect indicating the degree and kind of support provided by a teacher, was assessed through classroom observation. Direct instruction, which provides a very high degree of scaffolding, highly polarized motivational levels between boys and girls. With direct instruction a teacher provides step-by-step instructions at a detailed level (for example, “click this button,” “paint the frog green”). Direct instruction is particularly unappealing to girls. With less scaffolding, such as with guided discovery, a teacher employs a more inquiry-based approach that includes classroom discussion (such as “what should we do?” and “how can we do this?”). In guided discovery, the motivational levels of girls not only approached the motivational levels of boys but often exceeded it. In a number of classes, using guided discovery raised the motivational levels of girls to 100%. This is exciting because it suggests that broadening participation is not a question of difficult-to-change factors such as school affluence. Preconceived notions such as lower interest of girls in programming turned out to be unsubstantiated. Even in cases where most literature would suggest a gender effect, for instance if there were significantly fewer girls in a class, we found that the right level of scaffolding could raise the level of motivation in girls to be even higher than that of boys.

**Conclusion**

We believe we have found a systemic strategy for integrating CT education in middle schools in a way that exposes a large number of students and is appealing to girls as well as to underrepresented students. The Scalable Game Design project will continue as an NSF Computing Education for the 21st Century (CE21) project to advance a research based framework for broadening participation. We invite interested schools to participate.

**References**


Alexander Repenning (alex@cs.colorado.edu) is a computer science professor at the University of Colorado, a member of the Center for Lifelong Learning and Design at the University of Colorado, and the founder of AgentSheets Inc.

*Copyright held by author.*