Full Steam Ahead on CS-STEM Learning

May 24, 2011
Idit Harel Caperton, PhD, President & Founder, World Wide Workshop

The distinguished experts attending The Art of Science Learning Workshop in Washington in early April were onto something when they concluded that STEM learning—that is, education in science, technology, engineering, and mathematics—could well benefit from an infusion of art and design. Adding an A for art to STEM would give our technical and scientific education “some steam,” said my MIT colleague John Maeda, now president of the Rhode Island School of Design, by “grounding the bits and bytes in the physical world before us.” Paola Antonelli, Senior Design Curator at the Museum of Modern Art in New York City, added another element to the mix: “Art and design,” wrote Antonelli, “when used correctly, can integrate innovation into people’s lives.” STEAM and Innovation: They are and ought to be at the heart of both Computer Science learning and STEM education.

Constructionist Learning theory (full disclosure: I helped birth this theory with Seymour Papert at the MIT Media Lab in the mid-80s) has long pushed for this approach of combining computer science and engineering with the sciences and the arts. We put it into practice at the Media Lab itself as well as in the innovative work we pursued in schools worldwide two decades ago. The program was intentionally named “MAS” for Media Technology Arts and Sciences, and being highly interdisciplinary in nature, it combined architecture, media technology, arts, sciences, computation, design, music, and more. In fact, Constructionist theory and the STEAM it can provide were the subject of my PhD thesis book Children Designers and underlay the philosophy of the products I’ve been developing since.

Most recently, my organization, the World Wide Workshop, has applied this learning theory in Globaloria, the social learning network for game-based educational innovation that has been actively changing students’ academics, lives and career futures in several states. By imagining, drawing and building original videogames, Globaloria students have been demonstrating dramatically how art and
design and creative cognition can indeed ignite all kinds of STEM learning. Here are a few examples.

Globaloria students throughout the state of West Virginia participated this year in the 2nd Annual Globaloria STEM Games Competition. Participants have been researching various STEM topics, blogging about what they’ve learned, working in teams, producing video presentations, drawing paper prototypes, designing sample screens and graphics for game demos, and programming webgames that teach others about science issues or mathematic concepts. This year, 411 students signed up for the competition. 5 examples are presented here to illustrate the power of CS-STEAM. These students never programmed before. Click these links to play the games and review the learning process by clicking through the games demo slides, paper prototype videos, interactive screen designs, and presentation videos.

1. **Team Comatical Combat**: Play their game - *Elemental Elegance*. Visit the Team Studio (Design Process/Files/Prototype/Presentations Videos)

2. **Team Furyunleash22**: Play their game - *Paleo Quest*. Visit the Team Studio (Design Process/Files/Prototype/Presentations Videos)

3. **Team Energy Savers**: Play their game - *Sally’s Energy Ride*. Visit the Team Studio (Design Process/Files/Prototype/Presentations Videos)

4. **Team Trandon Berry**: Play their game - *Chemistry 101*. Visit the Team Studio (Design Process/Files/Prototype/Presentations Videos)

5. **Team XYZ**: Play their game - *Space Adventures*. Visit the Team Studio (Design Process/Files/Prototype/Presentations Videos)

*Constructionist CS-STEAM is at the heart of this.* It means not just learning by doing but learning by doing and by having to do it in the object’s own language. That is, if the object is to create a videogame about a social issue or scientific topic—environmental pollution perhaps, or space travel—students must not only learn about chemicals leaching into streams or how booster rockets power lift-offs; they must also master the mathematical abilities and technical skills needed to conceive, visualize, design, prototype, and program the videogame from start to finish. The programming language, animation scripting, computer technology, and Web 2.0 tools that are the means of enactment for the game creation are, in effect, the language through which students will read, visualize, write, and express the content of their chosen subject for their game.

Globaloria embodies those processes and skills in a comprehensive and customizable curriculum. In the classroom, working daily at their computers over a semester or two, the kids find themselves using mathematics, science, and technology tools as the building-blocks of their games. But since it’s all integrated, they learn the curriculum in the same way they learned their native language—experientially and creatively. Of course, as it turns out, as they gain this wonderfully
complex blend of digital literacies, their performance in “straight” math and science classes also improves.

In addition, Globaloria operates as a “design studio” or “digital collaborative.” It’s a social network in which kids work in individual design spaces but also have design spaces in which they work in teams—another critical skill for scientific thinking and career readiness. The teacher serves as a mediator and coach—as a conduit to a range of external resources available on the Globaloria network. In this transformed “blended” classroom, each student engages with the curriculum at his or her own pace and following his or her own bent, yet all come together as a team, both on-site and on-line, to analyze, troubleshoot, brainstorm, and solve big problems. Figuring out how to pull together all the game media elements, code, and other graphical components through multiple iterative design cycles is probably the hardest fun of all.

It’s the intellectual equivalent of kids getting their hands dirty with finger paints, brushes, glue, sand and clay, as they put their critical faculties to work doing research and thinking alone and together. They’re powering up their imaginations, strengthening their mental muscles of tinkering, debugging and critical analysis, as they master specific computing skills.

Globaloria is in action today engaging some 2000 students in 45 schools in three states. It will soon be in place in 60 schools in five states—middle schools, high schools, vocational and alternative education schools, charter schools. Significantly, we’ve targeted schools in technologically underserved and economically underprivileged communities, and the results have been pretty astonishing. In one of our newest schools, a charter school attended by kids for whom English is sometimes a difficult second language, kids who just a few months ago had never programmed a game or blogged, had no idea what Wikipedia is, and hadn’t a clue how to work on a wiki with a team are now doing it all. They are programming software, building their own computer games, studying via a network, and interacting remotely with experts who are 2000 miles away—with gusto and with increasing skill. Their math, reading, and writing performance is also improving.
For example, check out some voices from the field documentary video clips, to observe in particular how girls are included and empowered equally with the boys. One 6th-grade girl who appears in this last video clip even delved into a rather complex topic that was studied in her English class this year—the Holocaust—and drove her teammates to create the game they eventually called Survivor. They became obsessed with reading and researching about the Holocaust, and they designed impressive graphics, characters, and scenes as they focused on making meaning and building a game about the subject. It's a fascinating example of how students can grow their STEM knowledge and programming skills when given a chance to explore a topic of interest, even a tough one, and to develop a game around it. Their collaborative work on this is captured on Spongebob’s Crew Team Page.

In their recent article Can Arts Education Be a Savior to the Economy? Dr. Jim Denova, Vice President of the Benedum Foundation, and Gregg Behr, of The Grable Foundation, are also advocating for Science, Technology, Arts and Mathematics Education (STEAM) and holding up Globaloria as an exemplary model for the broader integration of arts across the educational spectrum.

Above all, Globaloria students of all ages in all grades are engaged in their own education in a way they have never been before. The chance for artistic expression plus bits and bytes plus physical, hands-on learning-by-doing have set their minds on fire. These kids have been turned on not just to digital design and the art of computational media technology, but also to visualizing and representing science knowledge in their games about climate change, molecular biology, and chemistry. Yes, they are learning how to make technology work, but in doing so, they are visualizing and re-imagining the world and explaining it to others in the language of the most compelling medium of this generation—game media. In short, they’re moving ahead full steam on CS-STEM learning and on preparing themselves for future careers and for full and active citizenship in a programmable and hyper-expressive, web-driven world.
Globaloria (www.Globaloria.org) is a social network for learning how to design and program webgames. It’s a rigorous, blended-learning platform, year-long academic curriculum, comprising programmable wikis and blogs, game programming tutorials, game-content resources, and virtual support. Students drive the design process, taking an original idea to final product. In playful learning, students are educated in both technical and computational skills and in content knowledge, thus preparing themselves for college-level STEM studies, 21st-century citizenship, and careers in the global knowledge economy. Globaloria can be found in 25 districts in West Virginia; in Austin, Texas; and Brooklyn, New York (and soon in California, Pennsylvania, and Florida). Globaloria applications and information: Amber@WorldWideWorkshop.org.

The World Wide Workshop (www.WorldWideWorkshop.org) inspires young people through social learning and the use of innovative computational media technologies. We enrich formal and non-formal learning systems that personalize learning opportunities—especially in economically and technologically disadvantaged communities. Our programs respond to President Obama’s national calls to action: “Educate to Innovate,” and “Change the Equation in STEM Education.” We work with forward-thinking leaders, corporations, universities, schools, and research centers to enrich public education systems with the latest technology and with innovative CS-STEM learning opportunities.

This paper was presented at the National Center for Women and IT 2011 Summit on Practices and Ideas to Revolutionize Computing. NCWIT SUMMIT is the pre-eminent destination for leading-edge research on curriculum, outreach, recruitment, retention, and advocacy across CS/IT pipelines. This annual event brings together leaders, change agents, and stakeholders to focus on research-driven practices that strengthen the computing workforce and cultivate technology innovation by increasing participation of girls and women.
The Globaloria Learning Process for Computational Thinking

1) **PARTICIPATE** in the Globaloria Social Learning Network to master content through computational inventiveness; work in teams, learn to solve programming problems and share computational knowledge publicly. Learn to collaborate onsite with classmates and educators, as well as virtually with students in other schools, and professional game makers and programmers.

2) **PUBLISH**: Learn how to present and publish designs, code, and games online.

3) **PROGRAM**: Write the code for your game in Flash Actionscript. Learn to program, test, and get help from experts using tutorials and virtual network for communication.

4) **PROTOTYPE**: Draw and videotape your game concept and test your prototype with users. Learn to use Flash to create an interactive demo that shows how the game will look and work.

5) **PLAY**: Play to discover what makes a great educational game. Learn about game mechanics, simulations, genres, and design principles. Get inspired!

6) **PLAN**: Decide who the audience is and what your game is going to do. Research your learning topics and learn your game content. Organize your ideas in a written plan. Keep adding to the plan as your research and design develops.

7) **PARTICIPATE** in the Globaloria Social Learning Network to master content through computational inventiveness; work in teams, learn to solve programming problems and share computational knowledge publicly. Learn to collaborate onsite with classmates and educators, as well as virtually with students in other schools, and professional game makers and programmers.

8) **PUBLISH**: Learn how to present and publish designs, code, and games online.

9) **PROGRAM**: Write the code for your game in Flash Actionscript. Learn to program, test, and get help from experts using tutorials and virtual network for communication.

10) **PROTOTYPE**: Draw and videotape your game concept and test your prototype with users. Learn to use Flash to create an interactive demo that shows how the game will look and work.

The 10 Design Principles for Cultivating Computational Inventiveness the Globaloria Way:

1) Learn by design of digital artifacts, functional, representational and educational games.
2) Master complex subjects by constructing pedagogical games for others.
3) Work on open-ended computational design tasks that focus on topics of choice.
4) Learn in a transparent studio setting (online/onsite) where work is built and shared.
5) Spend significant time on task (100 hours), in year-long project-based learning.
6) Have ample opportunities for social expression and discussion about game projects.
7) Have ample time for reflection about games, wikis, blogs, and representations.
8) Use programming and computational design tools as primary constructs of learning.
9) Use multiple modalities in the learning process (text, imagery, video, simulation).
10) Learn alongside educators (co-learning) and from experts (just-in-time learning).

The Key Constructionist Computational Thinking Abilities (Outcomes):

1) The ability to do invention, progression, completion of an original project; capability to program an educational game, wiki or simulation.
2) The ability to manage project-based learning in Web2.0 learning environments; capability to conduct and process complex project management (via programmable wiki systems).
3) The ability to produce original computational media; produce programming code, publish and distribute interactive purposeful digital media in social learning networks.