StarLogo TNG – Science in Student-Programmed Simulations

Eric Klopfer, MIT, Cambridge, MA, USA klopfer@mit.edu
Hal Scheintaub, Governor's Academy, Byfield, MA, USA hscheintaub@govsacademy.org

Abstract: StarLogo: The Next Generation (TNG), a new programming tool, was designed to rapidly engage students in game and simulation development in secondary school science and programming classes. TNG was introduced and used alongside established materials in three levels of high school physics. This pilot was designed to a) test the hypothesis that a game/simulation programming unit could harness the algorithmic thinking which is part of programming to provide alternate routes to understanding physics concepts, and b) explore the potential of motivating programming through game design/development in the context of traditional physics learning.

Models and Simulations in Science Education

Learning scientists are increasingly turning to video games as tools for learning (Gee, 2007). Simulations can motivate learners and provide ways for students to develop intuitive understandings of projectile motion (Jimoyiannis & Komis, 2003) and other abstract physics phenomena (Squire, 2007). However, because science practice involves the construction and validation as well as the application of scientific models, the authors share the belief that science instruction should include the making as well as the using of models (Hestenes, 2007).

Agent-based computer models are especially well suited for student inquiry and physics learning. The algorithmic thinking involved in programming such models emphasizes processes rather than facts (Cohen, 2007). Programming provides students with an alternative means of expression that is precise and compact (Sharin, 1993). Programming and algorithmic thinking offer alternative descriptions to complex phenomena that may be more accessible than algebraic descriptions to many students (diSessa, 2000), but questions remain as to a) how students become motivated to learn through programming and b) how to make the technical portions of programming accessible to a wide array of students and teachers.

StarLogo and StarLogo: The Next Generation (TNG)

Our efforts at integrating programming into schools builds on the Adventures in Modeling (AIM) program, which introduced model building to secondary school students and teachers using the older text-based version of StarLogo. Many AIM-trained teachers made simulations that became integrated into their classes (Scheintaub, 2007), but activities in which students built their own simulations were limited to only a few classrooms. Most teachers cited a lack of comfort with programming, concerns about syntax, and an inability to help in debugging, as primary reasons for excluding programming from their curriculum. Many also cited the "ramp up" time to learn programming as another barrier.

To promote a more widespread use of programming in schools, we designed StarLogo TNG. TNG provides two significant advances over text-based programming. First, the programming is done with graphical programming blocks instead of text-based commands. The blocks fit together only in syntactically sensible ways. This eliminates a significant source of program bugs that students encounter. TNG's second significant advance is a 3D representation of the agent world. Students can take the perspective of an individual agent in a realistic environment, which helps them see relationships between individual and system behaviors.

Learning Physics Through Programming

diSessa (2000) showed that fundamental physics concepts could be made accessible to students as early as sixth grade by using simple programming activities. We believed that at the high school level, programming could help students build a deep understanding of traditionally difficult physics concepts, so we introduced StarLogo TNG programming basics through a series of physics-based activities to three physics classes at a private school in the Boston metropolitan area. While students were not randomly assigned, three other comparable physics classes at the same school were used for comparison. Data was collected in the form of written assessments, lab reports, surveys and interviews.

The chosen unit focused on the topic of motion in two dimensions. Many strategies, including simulations, have been designed to help students understand that the vertical and horizontal motions of a projectile are independent of one another (Duran-Hutchings, 2004), but the concept of simultaneous but independent change remains difficult and frustrating to teach and learn. We hypothesized that by adding the programming of 2-D motion in a virtual world, to the usual mathematical analysis of that motion in the physical world, we could enhance understanding of this difficult concept. To be successful this approach would have to motivate students to challenge their pre-existing ideas, yet not place a great programming burden on the students or teacher. To accomplish this, students began the unit by building separate simple programming procedures that changed an agent's attribute like color, size or location. Procedures could be run separately or simultaneously, producing many humorous combinations of change. Students had little difficulty seeing that their agent's color change was independent of changes in its shape or its movement in the x or y-direction. To get realistic vertical motion students added a procedure for the negatively directed acceleration of free-fall to their independent x and y motions. To get an agent to jump over an obstacle, students had to separately build yet simultaneously execute a constant velocity, move-forward procedure and a vertical-jump procedure that employed acceleration.

Anecdotal evidence shows that students in the experimental classes transferred knowledge from their programming experience to new situations. In a lab report where students had to calculate the range of a projectile, one freshman wrote, "Programmers program motions independently in the virtual world; on the other hand physicists see the vertical and horizontal motion as independent of one another in the physical world. … I believe when an object is moving vertically it is independent and not interfered by horizontal motions. Our ball fell within our predicted value. This confirms our assumptions of vertical and horizontal motions being independent." The acquisition of new skills and confidence through programming is demonstrated in this anecdote. In a junior physics class students built swimmer-in-river simulations as part of a unit on vectors. The model has a swimmer with velocity, s, swimming at a given heading across a river of given width with current velocity, r. After building and playing with the simulation, students were assigned two-dimensional motion problems for homework. During a class discussion over the answer to a contested problem, one student went, unprompted, to the computer and opened his swimmer model. He plugged the variables of the problem into the code of the model and ran the model. As the swimmer reached the opposite shore he exclaimed, "I told you I was right!" When asked in an interview why he chose to use the computer rather than mathematical analysis to prove his point, he said. "This way you could ***see*** I was right." While we don't have comparative data, in another physics class that used StarLogo TNG, 75% of the surveyed students agreed with the statement that the StarLogo unit was more difficult than other units, while 100% of the students felt the unit was more rewarding; demonstrating the motivating potential of programming through game design/development in the context of a physics class.

Conclusions and Future Directions

The experimental classes finished their mechanics unit at the same time as the control classes and scored as well on standard assessments, while gaining the additional experience of model building, which is a critical component of modern science practice. These outcomes show that with TNG, programming and model building can become a part of a standard high school physics course and support expanded testing of this innovation in a more formal research setting.

References

Cohen, E. W.,& Kanim, S. E. (2007). Algebraic Difficulties in Physics. Retrieved November 16, 2007 from http://spacegrant.nmsu.edu/NMSU/2004/cohen.pdf

diSessa, A. (2000). Changing Minds: Computers, Learning and Literacy. MIT Press, Cambridge, MA, USA.

Duran-Hutchings, N. & Hutchings, J. (2004). Virtual Simulations of Projectile Motion and Solutions to a System of Parametric Equations, Retrieved November 19, 2007 from http://mathdl.maa.org/mathDL/3/?pa=content&sa=viewDocument&nodeId=1027&pf=1

Gee, P. (2007). What Video Games Have to Teach Us About Learning and Literacy, Palgrave Macmillan, NY.

Hestenes, D. (2007). Modeling Instruction in High School Physics, Chemistry, and Physical Science, Retrieved November 16, 2007 from http://modeling.asu.edu/modeling-HS.html

Jimoyiannis, A., & Komis, V. (2003). Computer Simulations in Physics Teaching and Learning: A Case Study on Students' Understanding of Trajectory Motion. *Computers & Education 36*, 183-204.

Scheintaub, H., Klopfer, E., Scheintaub, M., Rosenbaum, E. (2007). Complexity and Biology – Bringing Quanti-tative Science to Life Science Classrooms. in Linking Mathematics and Biology in High Classrooms, Fred Roberts, editor. in press.

Sharin, B., & diSessa A. A. (1993). Dynaturtle Revisited: Learning Physics Through Collaborative Design of a Computer Model, *Interactive Learning Environments, 3*, 91 – 118.

Squire, K., Barnett, M., Grant, J., Higginbotham, T (2007). Electromagnetism Supercharged! Learning Physics with Digital Simulation Games. Retrieved November 16, 2007 from http://www.educationarcade.org/files/articles/Supercharged/SuperchargedResearch.pdf