# NetTango: A Mash-Up of NetLogo and Tern

Uri Wilensky and Michael S. Horn, Northwestern University

***The Technologies:*** NetLogo is a popular agent-based modeling environment developed by Uri Wilensky and the Center for Connected Learning and Computer-Based Modeling at Northwestern University (Wilensky, 1999). We view agent-based modeling as a form of computational thinking in which elements of a complex system are represented by computational entities whose interactions result in emergent whole-system behaviors (Wilensky & Reisman, 2006). NetLogo models describe sets of rules that govern the behaviors of individual computational agents (e.g. sheep, wolves, and grass) in a system. As a model runs, interactions between agents, all concurrently acting out their rules, can help learners understand complex, real-world phenomena.

Tern is a tangible computer programming language that was developed by Michael Horn at Tufts University. Using Tern, children can create simple computer programs by constructing chains of wooden blocks. These blocks represent physical algorithmic structures that can be executed by a robot or some other computational agent. Tern is based on simple and reliable computer vision technology, making it inexpensive and practical for use in learning settings like classrooms and museums. Tern is featured in a computer programming and robotics exhibit at the Boston Museum of Science and as part of a curriculum unit designed for early elementary school classrooms (Horn, Crouser, & Bers, 2011).

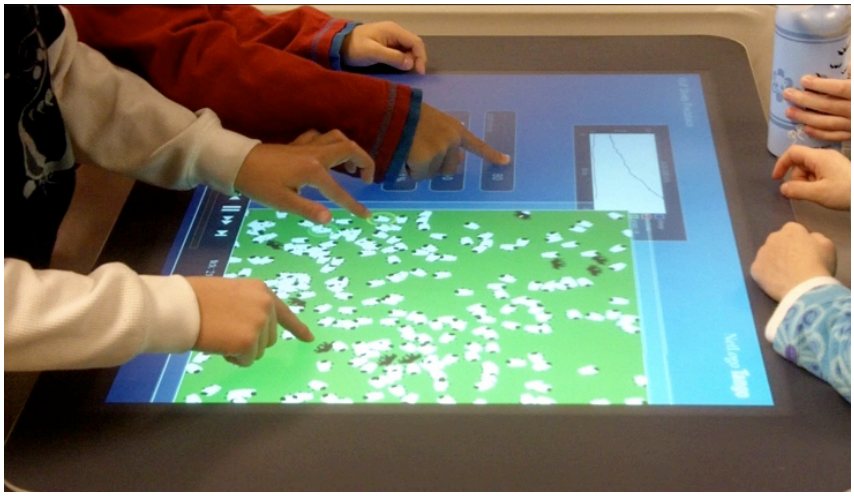

**Figure 1: Children exploring the wolf-sheep predation model in NetTango**

***The Vision:*** Wilensky and Horn teamed up when Horn joined the faculty at Northwestern University. The vision of the collaboration was simple—increase the accessibility of NetLogo for a younger audience and figure out a way to scaffold student learning, not only in terms of experimenting with NetLogo models, but also in terms of building their own models from scratch. With Tern we knew we had a good track record of encouraging a diverse audience to experiment with computer programming basics (Horn, Solovey, Crouser, and Jacob, 2009; Horn, Crouser, & Bers, 2011). Likewise, with NetLogo, we knew that we had a tool that could promote deep and meaningful engagement in a variety of content areas from chemistry to evolution. The trick, however, was making these two technologies work together to create a compelling learning environment. Our mashup, called NetTango, uses a multi-touch tabletop technology as the interaction platform to bridge the physical world of Tern with the virtual world of NetLogo (Olson, Leong, Wilensky, & Horn, 2011; Olson & Horn, 2011). In other words, the tabletop allows children to interact with NetLogo models using direct touch input while at the same time creating NetLogo programs using physical blocks tagged with computer vision fiducials.

NetTango is implemented in Java and currently runs on a Microsoft Surface. We use the NetLogo Java API as a modeling engine, allowing us to run existing NetLogo models with minimal

modification. Similar to NetLogo, NetTango provides the following interface components (Figure 2):

- A world window provides an animated visual depiction of the model as it is being simulated. Children can move and resize this window using standard multi-touch pinch and drag gestures.

- A control toolbar enables users to play, pause, rewind, and fast-forward simulations. There is also a scrub bar that users can drag back and forth along a playback buffer.

- A set of slider controls enable children to adjust model parameters (e.g. sheep reproduction rate).

- And plot windows display graphs of the model's execution (e.g. a plot of the wolf and sheep population numbers over time).

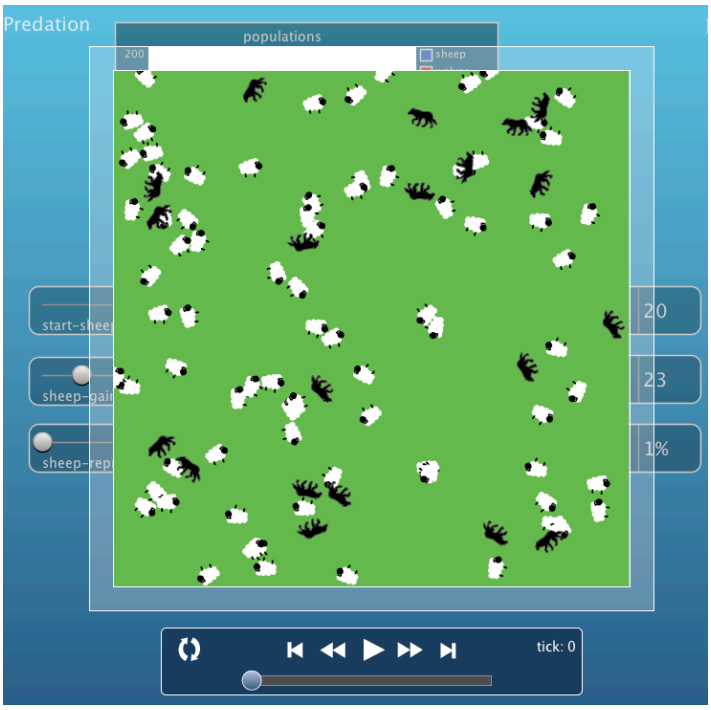


**Figure 2: NetTango provides a world window, parameter sliders, graphs, and a control toolbar.**

***The Future:*** This project is still in its infancy. We have begun testing prototypes with children and are exploring funding options. Fortunately, NetLogo and Tern speak a common language—the Java programming language. However, a much more important commonality is a shared vision and pedagogical orientation that we hope will result in a fruitful collaboration.

## References


Horn, M.S., Crouser, R.J., & Bers, M.U. (2011) Tangible Interaction and Learning: The Case for a Hybrid Approach. Personal and Ubiquitous Computing. DOI: 10.1007/s00779-011-0404-2 (online).

Horn, M.S., Solovey, E.T., Crouser, R.J., Jacob, R.J.K. (2009). Comparing the Use of Tangible and Graphical Programming Interfaces for Informal Science Education. In Proc. CHI'09 ACM Conference on Human Factors in Computing Systems, ACM Press.

Olson, I.C. & Horn, M. (2011). Modeling on the Table: Agent-Based Modeling in Elementary School with NetTango. In Proc. 10th International Conference on Interaction Design and Children (short paper), Ann Arbor, MI. June, 2011.

Olson, I.C., Leong, Z.A., Wilensky, U., and Horn, M.S. (2011). "It's just a toolbar!" Using Tangibles to Help Children Manage Conflict Around a Multi-Touch Tabletop. In Proc. International Conference on Tangible, Embedded, and Embodied Interaction (TEI'11), Funchal, Portugal. January 23-26, 2011.

Wilensky, U. (1999). NetLogo [computer software]. Evanston, IL: Center for Connected Learning and Computer-Based Modeling, Northwestern University. http://ccl.northwestern.edu/netlogo.

Wilensky, U., and Reisman, K. Thinking like a wolf, a sheep or a firefly: Learning biology through constructing and testing computational theories. Cognition and Instruction, 24(2), (2006), 171-209.