Running head: AGENT-BASED MODELING FOR PSYCHOLOGY

Piaget? Vygotsky? I'm Game!:

Agent-Based Modeling for Psychology Research

Dor Abrahamson and Uri Wilensky

The Center for Connected Learning and Computer-Based Modeling

Northwestern University

Contact Information:
Dor Abrahamson
Graduate School of Education
4649 Tolman Hall
University of California, Berkeley
Berkeley, CA 94720-1670
USA
TEL: +1 510 643 3599
FAX: +1 510 642 3769
e-mail: dor@berkeley.edu

Abstract

We discuss agent-based models as research tools for developmental and social psychology. "Agents" are computer-based entities, e.g., individual people or animals. To construct an agent-based model, the modeler assigns the agents real-world roles and rules, and then studies the model through conducting simulation experiments in which the agents follow their rules, and observes real-time data. Agent-based models are dynamic, expressive, and afford immediate feedback. Agent-based models are particularly useful for understanding *complex phenomena*, e.g., the dynamics of multiple individual learners interacting with their peers and with artifacts in their environment and the emergent group patterns arising over time from these multiple interactions. We describe an agent-based simulation that we designed as a "thought experiment" to shed light on the ongoing debate between two theories of learning, constructivism and social constructivism. When "running" the models, unexpected consequences arise, and this leads to successive refinement of theory. Practicing ABM in psychology research can potentially support the development of richer theoretical models that coordinate complementary perspectives as viable complements of an integrated explanatory structure.

Piaget? Vygotsky? I'm Game!:

Agent-Based Modeling for Psychology Research

>"[T]he social fact is for us a fact to be explained, not to be invoked as an extra-psychological factor."
>
> Piaget (1962/1951, p. 4)
>
>"The development of thought is to Piaget, a story of the gradual socialization of deeply intimate, personal, autistic mental states. Even social speech is represented as following, not preceding, egocentric speech."
>
> Vygotsky (1962/1934, p. 18)

In this paper, we present a new methodology for conducting research on psychology theory, *agent-based modeling*. Agent-based modeling (hence ABM) has been increasingly used by natural scientists to study a wide range of phenomena such as the interactions of species in an ecosystem, the interactions of molecules in a chemical reaction, the percolation of oil through a substrate, and the food gathering behavior of social insects (Sole & Goodwin, 2000; Bonabeau, Dorigo, & Théraulaz, 1999; Troisi, Wong & Ratner, 2005; Wilensky & Reisman, 1998, 2006).

What all these studies have in common that make them particularly conducive to ABM treatment is that the focal phenomenon in question can be construed as a pattern resulting from the cumulative activity of many elements. For example, the phenomenon of pressure, e.g., in a gas chamber, is explained as the cumulative collisions of myriad particles with the walls of the chamber. Such phenomena, which lend themselves to two or more layers of description, e.g., particle collisions are the "micro" events, and pressure is the "macro" event, have been termed *complex*, and are collectively studied in a relatively young interdisciplinary field called *complex systems* or *complexity studies*

(Axelrod, 1999; Bar-Yam, 1997; Holland, 1995; Kauffman, 1995; Wimsatt, 1986, cited in Clark, 1997, p. 114). Typical of complex phenomena is that the cumulative ('aggregate') patterns or behaviors at the macro level are not premeditated or directly actuated by any of the "lower-level" micro elements. For example, flocking birds do not intend to construct an arrow-shaped structure. Rather, each element ("agent") follows its "local" rules, and the overall pattern arises as epiphenomenal to these multiple local behaviors—the overall pattern *emerges*. The agents can be instantiated in the form of a computer program that specifies their rule-based behaviors. ABM is thus particularly powerful for studying complex phenomena, because once the modeler assigns agents their local rules, the modeler can set these virtual agents into motion and watch for any overall patterns that arise from the agents' interactions. For example, the modeler might assign a group of virtual birds a set of rules and then watch their interactions to see whether typical flock structures emerge (Reynolds, 1987; see also Wilensky, 1998).

Whereas initially complex-systems methods and perspectives arose from the natural sciences, complexity, emergence, and micro and macro levels of description of phenomena are all highly relevant to research in the *social* sciences. Indeed, the recent decades have seen a surge in social-science studies employing ABM (Epstein & Axtell, 1996; Diermeier, 2000; Axelrod, 1997). In this paper we focus on the potential of ABM as research tools for formulating and critiquing social-science theory. We do so by looking at a specific example, the debate between Piagetian and Vygotskiian accounts of learning, and we introduce our main result as an instance of the methodology we are advocating. We argue that ABM has potential to contribute to the advancement of theory in at least three major ways that we illustrate in this paper: (a) explicitizing—ABM computational environments demand an exacting level of clarity and specificity in expressing a theoretical model and provide the tools, structures, and standard practices to achieve this high level; (b) emergence—the computational power of ABM enables the researcher to mobilize an otherwise static list of conjectured behaviors and witness any group-level patterns that may

enfold through multiple interactions between the agents who implement these conjectured behaviors; and (c) intra/inter-disciplinary collaboration—the *lingua franca* of ABM enables researchers who otherwise use different frameworks, terminology, and methodologies to understand and critique each others' theory and even challenge or improve the theory by modifying and/or extending the computational procedures that underlie the model. In fact, we invite other researchers to explore the advantages of including ABM models with their theories. To demonstrate the plausibility of such model-based discourse, we welcome theory-of-learning researchers to challenge the model that we present in this paper.[1]

What Is an Agent-Based Model?

The agent-based model is the researcher's idealized approximation of 'how things work in the world.' "Agents" are computer-based entities, e.g., "people." The modeler creates the agents that are conjectured to be instrumental in the real-world phenomenon under study and assigns these agents rules that express the modeler's conjecture as to real-world behaviors. The modeler also creates a virtual environment where these agents operate. Next the modeler conducts simulation experiments in which the agents play out their rules within the environment and in response to each other. Running these simulations is *not* just an animated enhancement of the programmed rules within a visual medium. The dynamic interactivity of these agents is such that it is often very difficult or even impossible to predict emergent outcomes from the agents' individual rules and the environmental attributes. Moreover, the random values inherent in each simulation make for different outcomes each time a specific model is run. Thus, the craft of ABM involves "informed fiddling": the modeler iteratively explores within a vast range of parameters, including the environmental settings and the logic of the agents' behaviors, until a simulation is run that yields patterns that appear familiar from observations of real-world data. This virtual-to-real similarity does not constitute adequate proof of the hypothesized rules, yet it renders the

---

[1] After presenting a version of this paper at the Jean Piaget conference in June 2005, some researchers did respond to this challenge (e.g., Levin & Cole, 2006)

rules plausible and sets an agenda for attempting to validate the rules through systematic exploration of the model space as well as through complementary methodologies.

The proposed computer-based methodology is a new important research tool in the complement of tools available to social-science researchers. Specifically, ABM offers a methodological framework with the following unique set of attributes: (a) ABM shares the rigor of laboratory experiments and amplifies this rigor; (b) In ABM, the observer is literately "invisible" to the participants in ways that cannot be achieved in ethnographical interventions; (c) ABM enables "to see eternity in an hour," i.e., to rapidly conduct simulations of longitudinal studies and conversely, ABM can break down a brief phenomenon into a sequence of discrete actions; (d) ABM flexibly integrates the "micro" perspective, that is the thinking and behavior of individual participants as collected from, for example, clinical interviews, with the "macro" perspective that is the overall large-scale patterns in group behavior that is typically collected from studies involving thousands of participants; and (e) ABM is a risk-free environment—researchers can investigate aspects of human behavior under conditions that would compromise the safety and sentiments of human subjects, e.g., in studying group behaviors that emerge in the emergency evacuation of a building.

The computational "language" in which researchers express their hypotheses, e.g., "IF–THEN" statements, demands of the researchers high precision, explicitness, and specificity in deploying their conjectured theoretical models. Thus, peers who are sufficiently proficient in the modeling language can evaluate the coherence of the computer-based model by examining the underlying modeling code. Such peers can modify, revise, or extend the model to investigate alternative explanatory structures. Thus, ABM is a form of pellucid scholarly rhetoric. In particular, ABM has the capacity to enable scholars operating within different methodologies to communicate in a *lingua franca.*[2] In this paper, we

---

[2] For the past two decades, Wilensky has been engaged in developing NetLogo (Wilensky, 1999) an ABM environment specifically designed to be easy to learn and to "read" that can serve as such a *lingua franca*.

demonstrate the utility of ABM methodology to illuminate a particular controversy in theory of learning, constructivism versus socio-constructivism.

What's New About This (or, Haven't Computer Models Been Around for Quite a While?)

It is said that a picture is worth a thousand words. Paraphrasing this adage, Goldenberg, Lewis, and O'Keefe (1992) have argued that a moving picture is worth a thousand pictures. Motion has often been cited as an advantage of computational representations, along with computational speed, data logging and accessing, and interlinked representations. We wish to further paraphrase that an agent-based model is worth a thousand moving pictures. More precisely, we wish to emphasize that ABM bears critical features that make it a much more powerful learning tool as compared to animations or visualizations.

ABM dynamic visualizations are open ended. Typically, each ABM "run" begins with a set of agents bearing random values, so that the initial conditions usually differ from run to run. Moreover, a measure of randomness can be incorporated into the execution of agent procedures, resulting in end states that may differ in details from run to run even if these runs share initial conditions. Yet the randomness inherent in the setting and running of the simulations only contribute to the psychological effect of witnessing overall emergent patterns that may be shared across these superficially different simulations. Perhaps most important to researchers, though, is the malleability of the ABM simulation, a malleability that enables researchers to continuously explore the implications of modifying their theoretical model. The malleability of ABM simulations is *technically* enabled by the simplicity of the modeling "code" in which the agent rules are expressed (see later sections). Yet the very choice to model phenomena from the agent perspective (micro) and not, as has been the general rule in the sciences, from a global "aggregate" perspective (macro), is associated with the unique epistemic affordances of agent-based explanatory mechanisms. Both agent-based and aggregate models provide tools for articulating explanatory coherence—ABM are expressed in rules (procedures) and aggregate

models are expressed in differential equations. Yet whereas agent-based rules are generative, aggregate rules are summative. That is, agent-based models express *a priori* the 'doing' underlying the phenomena, whereas aggregate models capture *a posteriori* the global relations between aggregates of agents (without explicitly providing insight into the causality underlying these global relations). Thus, ABM and aggregate modeling are not commensurate or interchangeable epistemic forms for thinking about phenomena—they are radically different. Ultimately, both approaches are useful and can be seen as complementary to scientific practice (Wilensky & Stroup, 2003; Wilensky & Abrahamson, 2006). Whereas we focus on the phenomenon of learning, as a case study, this type of experimentation, we argue, can illuminate many issues facing developmental and social psychologists as well as education researchers.

In the remainder of this paper, we begin by explaining our choice of the phenomenon of 'learning' as a case study for demonstrating the potential of ABM, we continue by further explaining complexity studies and agent-based modeling, and then we present the particular model we built for this study, including findings from running this model and interpretations of these findings, which lead to our conclusions

Background

Recently, at an international conference, an author was criticized for espousing both *constructivism* and *social-constructivism* as theoretical frameworks for conducting research of student learning, as though these twain shall never meet, let alone collaborate. Despite efforts to reconcile these schools of thought (e.g., Tudge & Rogoff, 1989; Cole & Wertsch, 2002; Fuson & Abrahamson, 2005), by and large they are still considered rival rather than complementary. The rift between these schools is most evident in the ongoing efforts of Learning Sciences scholars to create new paradigms that encompass both schools. For example, Greeno (2005) explains that his *situative perspective*

"…builds on and synthesizes two large research programs in the study of human behavior….*cognitive science* [that] focuses on individuals, although it occasionally considers social interactions as context of individual cognition and learning….[and] *interactional studies* [that] focuses on performance of groups of individuals engaged in joint action with material and informational systems in their environment….[Yet] neither [of these research programs], by itself, can explain learning."

Why is this dichotomy still prevalent? It could be that the lingering rivalry between cognitive and interactional perspectives on learning is a vestige of college and graduate-school introductory courses that neatly compartmentalize the literature into juxtaposed schools of thought—a juxtaposition that traces the historical evolution of research on learning by foregrounding distinctions between schools but does not provide tools for examining the complementarity of these paradigms. Also, some scholars may inadvertently disseminate and, thus, perpetuate an apparent schism in the field of research on learning, because they are still entrenched in paradigms they themselves were schooled in (Kuhn, 1962). So we are faced with what appears to be an intellectual impasse, an impasse that perpetuates an historical schism that may be ill founded to begin with. It appears as though we haven't the methodological wherewithal to ponder this juxtaposition in ways that we would be comfortable to endorse as experimentally rigorous, theoretically coherent, and rhetorically compelling. We submit that holding on to such a position—that one of the schools is essentially more revealing of patterns and mechanisms of learning—limits the scope of effective research and educational design, especially design-research for classroom learning.

It could be that the two "camps" do not interact, because they lack a common "campground." That is, fruitful discourse between Piagetians and Vygotskiians is constrained, because they operate in parallel paradigms including separate perspectives, methodologies, and terminology.

The challenge and consequent rarity of effective Piagetian–Vygotskiian explanatory models may thus be addressed by forms of modeling that are adequate for both perspectives. We now further discuss agent-based modeling, which we suggest as a methodology that bridges the two theoretical perspectives synergistically.

Agent-Based Modeling (ABM)

"When I observed phenomena in the laboratory that I did not understand, I would also ask questions as if interrogating myself: 'Why would I do that if I were a virus or a cancer cell, or the immune system.' Before long, this internal dialogue became second nature to me; I found that my mind worked this way all the time" (Salk, 1983, p. 7).

This paper proposes the viability of agent-based modeling (ABM) environments (e.g., 'NetLogo,' Wilensky, 1999a; 'Swarm,' Langton & Burkhardt, 1997; 'Repast,' Collier & Sallach, 2001) as research tools for articulating and examining hypotheses (Wilensky, 2001; Wilensky & Reisman, 1998, 2006) in the domains of developmental and social psychology, such as in the research of learning. ABM of human behavior is a growing research practice that has shed light on complex dynamic phenomena (e.g., Kauffman, 1995; Holland, 1995) such as residential segregation (Schelling, 1971), wealth distribution (Epstein & Axtell, 1996), negotiation (Axelrod, 1997), and organizational change (Axelrod & Cohen, 1999). Learning, too, is complex—it involves multiple agents interacting with artifacts and peers—so research of learning may, too, benefit from ABM. To demonstrate the utility of ABM as "thinking tools" for psychology research, we have designed and implemented in NetLogo the "I'm Game!" simulation, in which *agents*, computer-based entities, "learn" through playing either as individuals, social interactors, or both.

The NetLogo environment (Wilensky, 1999a) was designed with the explicit objective that it provide a "low threshold, high ceiling" for students and researchers alike to engage in agent-based modeling (Wilensky, 1999b; Tisue & Wilensky, 2004). The vision is that building simulations will

become a common practice of natural and social sciences scholars investigating complex phenomena—the scholars themselves, and not hired programmers, build, run, and interpret the simulations. To these ends, the NetLogo "language" has been developed so as to be accessible—easy to write, read, and modify, thus making NetLogo very much distinct form common general-purpose programming languages like Java and C++.

In order to introduce ABM further, we will examine, in later sections, the NetLogo code that underlies the "I'm Game!" simulation and expresses the theoretical assumptions of the modeler (on how social entities are expected to behave under particular conditions). The entire NetLogo code for the "I'm Game!" model occupies about five pages of sparse programming "text," or a total of just over 820 "words" and symbols (under 4000 characters), of which 455 are the core code, and the rest is auxiliary to the simulation itself—various setup, monitoring, and graphing procedures (see Appendix A). Thus, a caveat is, perhaps, due at this point, for readers with little or no experience with computer-based simulations who are concerned with the brevity of the code vis-à-vis their bookshelves-full of relevant literature. First of all, the model does not purport to contain interpretations of behaviors but only to support an inquiry into these behaviors. Secondly, it is the nature of NetLogo-type code that complicated syntactical structures can be condensed into short "sentences." Thirdly, the nested structure of computer procedures, combined with a defined parameter space and a randomness generator, allows for a limited number of specifications to generate a panoply of individual scenarios (we can "collapse" the combinatorial space of possible events). That is, the dynamics and randomness-capacity of the simulations is specifically geared so as to allow the narrative of various case studies to emerge through running the simulation with a "cast" of numerous agents who are operating in parallel. In other words, enfolded or embedded within the condensed code is the potentiality of literally endless "cases," a potentiality that bears out when the program is activated and the data are automatically collected and measured.

The construct of 'emergence' implicates both a limitation of mental computation, i.e., that certain phenomena surprise us, because we do not have the computational wherewithal to infer them from the 'givens' (Wilensky, 1993, 1997; Wolfram, 2002)—and the utility of 'thinking *with*' technology (Norman, 1991; Salomon, Perkins, & Globerson, 1991; Wilensky, 1999, 2002). For example, it is difficult to anticipate the outcome in the following scenario:

A termite colony is foraging in a landscape strewn with randomly distributed woodchips. Each termite individually follows the following rule: "Walk forward in some random orientation. If you come upon a chip, pick it up and carry it in a random orientation. If you come upon another chip, search for an open space and place there the chip you are carrying" (Resnick, 1994; Wilensky, 1998).

When hundreds of simulated termites each follow these rules, the stable end result is that all the chips are collected into a single large pile.

The apparent mental challenge in anticipating the 'termites' phenomenon is that the output of each step constitutes input for the next step—the mind is apparently overwhelmed by the computational demands of iteratively tracking these myriad parallel events. For a novice at complexity, there are no apparent heuristics to inform a shortcut or leap from the 'start' to the 'end.' Thus the epistemological position that the ultimate outcome 'emerged,' and thus the need for multi-agent modeling-and-simulation environments to connect from the 'micro' perspective on agent rules to the 'macro' perspective on global patterns (Wilensky, 1997).

In summary, ABM shares with classical experimental designs the logic that particular aspects of a studied phenomenon are foregrounded for observation, measurement, data collection, and analysis. However, ABM is distinct from classical designs in that the experiment is simulated—it does not actually happen "out there"—not even in a classical laboratory. In this sense, ABM is more like a thought experiment, a computer-enhanced *Gedankenexperiment*—it is a means of exploring the eventualities of beginning from a set of well defined simple rules and allowing the experimental "subjects" to act out these rules with a measure of randomness, possibly giving rise to surprising group

patterns. Ultimately, results of these experiments must be interpreted with caution. Even if group patterns emerge that appear convincingly similar to behaviors "in the world," one should maintain salubrious skepticism, in case critical factors were not included in the model. But, then again, such tenuousness is a feature of any post-Popperian scientific research.[3]

Learning Through Modeling

Modeling is not only for professional scientists. Researchers of learning concerned with the design of learning environments have pointed to the efficacy of building models as a form of learning (Hmelo-Silver & Pfeffer, 2004; Lesh & Doerr, 2003; Resnick, 199x; Wilensky, 1997; 1999b; Wilensky & Reisman, 1998, 2006). A process of modeling may consist of foregrounding elements of phenomena under inquiry, including these elements' properties and behaviors, and accounting for these phenomena, e.g., by implementing explanatory mechanisms of causality. Through modeling, students (in the broad sense of 'learners' of all ages, walks of life, and expertise) concretize and articulate their understanding of content in the form of some expressive artifact, such as a diagram, a narrative, a mathematical equation, or a computer procedure. Students can then display these artifacts in a public forum, explain their intended interpretation of these artifacts (i.e., what they are modeling), and negotiate with their peers both the adequacy of the model (that it is modeling what it purports to model) and their understanding of the phenomena as expressed in the model. Students discuss their understanding and modify it based on their own insight as well as on their peers' response.

One form of modeling that has been receiving increased attention is computer-based modeling. In computational environments that provide feedback to the user, students' conceptual models and virtual models are mutually informative. That is, students' reasoning and their computer procedures

---

[3] "A good scientific model….should have a certain 'open texture', giving scope for further exploration: it should not merely seek to summarise what is known. Then it can be a research tool as well as a teaching aid. The model itself can be studied for *its* properties and this may suggest new possibilities (or impossibilities) and raise new questions…. Of course you should never *believe* models, they only have an 'as if' status and they are likely to let you down any time" (Cairns–Smith, 1996, p. 46-7).

develop reciprocally: students "debug their own thinking"—that is, detect and correct their own misconceptions—through debugging the program procedures (Papert, 1980). For instance, students who attempt to cause a computer-based agent to bounce across the interface, realize that a "go up, go left, go down" procedure does not result in bouncing as they know it "in the world." Through this breakdown experience, bouncing "announces itself afresh," if to paraphrase Heidegger (1962). Specifically, students become aware of the arc motion of the desired ballistic trajectory, and may be, thus, stimulated to formulate an improved procedure that results in the desired arc. Thus, through the very act of implementing within a computational environment their model of a phenomenon, learners have opportunities to develop more sophisticated understanding of the phenomenon.

Agent-based modeling is particularly suitable for learning complex phenomena (Wilensky, 1999b, 2002; Wilensky & Resnick, 1999; Jacobson & Wilensky, 2005; Resnick, 1994). One reason complex phenomena are difficult to understand is that they can be perceived from two or more levels of coherency. For instance, a flock of birds flying in an arrow-shaped formation can be just that—a flock (the "macro level")—or it could be regarded as a collection of individual birds each following some rules (the "micro level"). Both perspectives on the flock, the micro and the macro, are necessary to understand the flocking phenomenon, because it is through the rule-based interactions of the individual birds that the flock form emerges as a recognizable entity in the world. Therefore, a computer-based environment for modeling a flock and simulating its emergence requires tools for programming individual agents, e.g., the birds, parallel processing capabilities for the birds to operate simultaneously and interact, and functions for quantifying aggregate (macro) properties, such as the total number of birds or their mean orientation. A group of learners is a complex phenomenon, too, and so we have used the NetLogo multi-agent modeling-and-simulation environment to create the model that accompanies this paper.

How do flocks relate to students? In creating the "I'm Game!" model, we sought to understand learning as an individual process that occurs within a social context, in which individuals both contribute to and, in turn, are influenced by this context of fellow learners. At the same time, we were interested in measuring the impact of rules we assign to individual students on the learning of the group.[4] That is, we sought to examine any emergent group-level phenomena that may result from interactions between individual learners.

In summary, modeling for learning affords much more than capturing a *post facto* product of a completed inquiry process. On the contrary, building the model is intrinsically intertwined in the learning process itself. The model is "an object to think with" (Papert, 1980). Computer-based modeling environments, in particular, bear constraints that scaffold practices with strong parallels to theoretical modeling: specificity and consistency in terminology, rigor in articulating relations between elements (rules, topology, temporality), definitions of the parameter space that is informed by implicit hypotheses, and coherence, which is tied to the esthetics and elegance of the code and the interface (a latter-day Occam's Razor). Also, computation-based simulations can stimulate scholarly discourse by facilitating a sharing of methodology, experimentation, and data at a level of explicitness that enables thorough scrutiny and precise replicability (Wilensky, 2000; Tisue & Wilensky, 2004).

We have discussed researchers' increasing use of ABM for revealing patterns is complex social phenomena, and we have discussed the role of modeling in learning. In the remainder of the paper, we examine a case study of modeling a complex psychological phenomenon, 'learning,' and discuss results from operating this simulation. In particular, we will focus on how building the "I'm Game!" model of 'learning through playing' transposed the negotiation between two theoretical camps to a shared

---

[4] By "group learning" we are not referring to collaboration-oriented distributed-cognition literature (e.g., Hutchins, 1995) but to indices that capture aggregate aspects of a collection of individual learners' independent behaviors. That is, in building our model, we did not intend to feature *intentional* collaboration between agents with a common objective.

campground where, rather than translating between camp-specific terminology, we adopted the computer-based *lingua franca* of NetLogo code to express the theoretical models of both camps.

Agent-Based Modeling: A Case of Modeling 'Learning Through Playing'

The "I'm Game!" model of 'learning through playing" was created with the following objectives: (1) to demonstrate the viability of agent-based modeling (ABM) for examining socio/developmental-psychological phenomena; (2) to illustrate the potential of ABM as a platform enabling discourse and collaboration between psychologists with different theoretical commitments; and, specifically, (3) to visualize the complementarity of Piagetian and Vygotskiian explanations of how people learn.[5]

We strove to create a simulation that would be simple enough to carve the theoretical models at the joints yet would still allow a meaningful distinction between them. This *reducto* exercise proved useful as a means toward implementing the theoretical models in the NetLogo code, in the form of simple and concise procedures.

To model human learning in social contexts, we chose the context of a group of children at play, because both Piaget (1962) and Vygotsky (1962) studied relations between playing and cognitive development.[6] Also, games easily lend themselves to proceduralization in computer code as a set of

---

[5] To run this experiment, go to http://ccl.northwestern.edu/research/conferences/JPS2005/jps2005.html, an on-line web-page accompanying this paper, where you will find an "applet" (an interactive simulation embedded within an html page). The model itself is available from the authors.

[6] Piaget saw games as "ludic activity of the socialized being" (1962, p. 142)—assimilation-based forms of practicing the execution of schemas (see also Huizinga, 1955). "There are practice games, symbolic games, and games with rules, while constructional games constitute the transition from all three to adapted behaviors" (1962, p. 110). Rule-based games, specifically, result from, "collective organization of ludic activities" (1962, p. 113). For the main, Piaget studied the function of play vis-à-vis his stages of cognitive development. Vygotsky, too, saw games as crucial for development—although occupying a limited part of the child's time, play is, "the highest level of pre-school development" (1978, p. 102). For instance, Vygotsky regarded, "make-believe play as a major contributor to the development of written language—a system of second-order symbolism" (1978, p. 111). However, Vygotsky interpreted humans' urge to play within a broader psycho–social web of practices, for instance as a substitute to unfulfilled or unrealizable desires. Also, he stresses the pervasiveness of imagination and rules in play: in early play, tacit rules, e.g., "sisterhood," are made explicit; and, "Just as the imaginary situation has to

conventional rules that agents follow. That is, we could have chosen the activity context of "Making

Pasta" (recipe as rule-based), "Buying a Toy" (some algorithm that considers affective and financial

input), or even "Making Music Together" (modifying pitch and tempo based on ongoing feedback from

peers). But a context in which a group of participants are engaging in an *explicitly* ludic activity enables

us, the modelers, to define unequivocally and, hopefully, incontestably, the specific objectives and

underlying skills that participants need to develop in order to engage successfully in the organized

activity. That is, by choosing a context in which skills are *commonly* discussed in terms of procedures

(e.g., "If you see that someone is trying to steal base, then you should…etc"), we eschew debating the

potentially controversial stance that any human activity is given to definition as a set of procedures that

can be implemented as computer code. Finally, we chose a game involving marbles as homage to Piaget

for his studies in these contexts (e.g., Piaget, 1971).[7]

The design problem for this model-based thought experiment was to create a single environment

in which we could simulate both "Piagetian" and "Vygotskiian" learning. In the interest of

foregrounding the theoretical *differences* between these two perspectives, we focused on the differential

emphases they put on the contribution of the social milieu to individual learning. Thus, we wanted to

visualize and measure incremental improvement in learners' performance in the marbles game under

two conditions: (a) "Piagetian"—players interact with objects in their environment, accommodating

their schemas based on feedback; (b) "Vygotskiian"—players learn through imitation by participating in

an organized activity that includes others who are more experienced in performing a certain target skill;

and (c) "Piagetian–Vygotskiian"—learners learn both from their own performance and from the

---

contain rules of behavior, so every game with rules contains an imaginary situation" (1978, p. 75), even
chess. He summarizes that, "The development from games with an overt imaginary situation and covert
rules to games with overt rules and a covert imaginary situation outlines the evolution of children's
play" (1978, p. 96). The motivations for the gradual prominence of progressively rigid rules are
children's interest (simply running becomes boring) and a social caveat to distinguish more clearly
between work and play coupled with the regulatory assimilation of a rule-based social ecology.
[7] Piaget reports on studies of children's understanding and attitude towards the rules of the game of
marbles. Our simulations focus not on learning rules but on learning to perform better in the game.

performance of others. Clearly, such typifications of "Piagetian" and "Vygotskiian" learning are gross caricatures of these theoretical models. Yet the simple *agent* rules stemming from these typifications may be sufficient to generate data revealing interesting behavioral patterns at the *group* level. One could always build further from these rules by complexifying the agents' rules as well as the contexts of the simulated "world."

It is possible to simulate two different types of behavior—"Piagetian" or "Vygotskiian"—within the same model. This is achieved by defining separate sets of rules for the agents and running the simulation either under one of these conditions or the other. In fact, as we will detail in a later section, it is possible to simulate a combination of these two behaviors—"Piagetian *and* Vygotskiian"—by creating an option in which both types of behaviors are active and some selection procedure then governs a resolution as to which output shapes the subsequent behavior of the agents. For instance, the selection could be utilitarian—the output is selected on the basis of its goodness, that is, the degree to which it helps the agent achieve a specified objective.

[Insert Figure 1 about here]

In the interest of enhancing the "legibility" of the simulation's computer interface, we sought a marbles game in which a learner's skill can be visually indexed even without resorting to quantitative metrics that usually appear in interface display windows in the form of numerical values. We eventually chose to model a game (see Figure 1) in which contestants all stand behind a line and each throws a marble, trying to land it as close as possible to a target line some "yards" away (30 NetLogo 'units' away). This game, we find, enables – at a glimpse – to evaluate how well players are doing, both as individuals (distance to the target line) and as a group (mean distance to the line or density). We will now explain the game in more detail, focusing on how we implemented, measured, and displayed

Piagetian and Vygotskiian learning.[8] We will then turn to some of the core computer procedures underlying the simulation, so as to demonstrate how the NetLogo computational medium enables an implementation of theoretical models (see Appendix A for the entire NetLogo code of this model).

Rules of the "I'm Game!" Agent-Based Model of Learning in Social Contexts

Players stand in a row (see Figure 1a, above). They each roll a marble at a target line. Some players undershoot the line, some overshoot it (see Figure 1b). Players collect their marbles, adjust the force of their roll, and, on a subsequent trial (Figure 1c), improve on their first trial—they have "learned" as individuals. Gradually, the group converges on the target line (see Figure 1d, 1e, and 1f, for three later attempts).

[Insert Figure 2 about here]

Figure 2 shows the entire interface of the "I'm Game!" interactive simulation, including all controls and displays. We simulated four learning strategies: (a) "Random"—a control condition, in which players' achievement does not inform their performance on subsequent attempts, unless they are exactly on target; (b) "Piagetian"—players learn only from their own past attempts; (c) "Vygotskiian"— players learn only by watching other players nearby, not from their own attempts; and (d) "Piagetian– Vygotskiian"—players learn from both their own and a neighbor's performance. In Figure 2 we see results from having run the simulation under the four conditions and over about eighty experimental runs, each consisting of thirty attempts. The graph in the center, "Average Distance," shows the group mean distance from the target line by strategy, and the histogram on the right, "Strategy Averages," records the group's cumulative mean performance by strategy. (Note that attempts are measured by the absolute distance of a marble from the target line, so in both the graph and the histogram, lower scores are associated with better, or more skilled, performance.)

---

[8] When we drop the scare quotes from "Piagetian" and "Vygotskiian," we still bear in mind that the model is a caricature of the theories.

To make the learning process more realistic, we implemented a random-error parameter that introduces "noise" into the learning process (in Figure 2, it is set to 12, in the slider that is third from bottom, on the left side of the interface). Also, we implemented the "#-Vygotskiian-neighbors" parameter (in Figure 2, "#-Vygotskiian-neighbors" is set to 10, in the second-from-bottom slider, on the left) that controls the number of neighbors each player observes in the Vygotskiian strategy (the player's subgroup). Finally, we incorporated a "ZPD" (*zone of proximal development*) slider (see in the center of the button-and-slider section). This variable (here, set to 10) limits the range of performances that players can imitate. For instance, a player can imitate a neighbor only if the neighbor's score was better by 10 units or less, as compared to the player. So (see Figure 2, the picture with the "marbles"), the bottom player cannot imitate the player three rows above it, because the difference between their respective performances is larger than the current value of ZPD, but it can imitate the players one and two rows above it, who are well within its ZPD.

Note that the learning process involves "feedback loops." That is, a player's learning—the individual "output" of a single attempt—constitutes "input" for the subsequent attempt. In the "Piagetian" condition, this is a player-specific internal loop, and in the "Vygotskiian" condition one person's output may be another person's input on a subsequent attempt, and so on. Note also that over the course of a "Piagetian–Vygotskiian" run of the simulation, players might learn on one attempt from their own performance and on another—from the performance of a neighbor. Both sources of information are simultaneously available for each player to act upon.

Computational Procedures That Implement a Theoretical Model

An introduction to ABM would not be complete without a glimpse into the code that underlies the agents' rule-based behaviors. The semantics and syntax of the NetLogo programming code is developed with particular attention to users' expectations, coming from natural language (see Tisue & Wilensky, 2004). To the extent possible, code "primitives" (the basic commands that the program

"knows") are simple nouns, verbs, quantifiers, and prepositions, such as "turtle," "breed," "forward," "back," "jump," "count," "beep," "any?," "nobody," "-from," "-at," or "clear-graphics." Users "teach" the program new words (for actions and variables) for particular models they are building. For example, a variable in the "I'm Game!" model is "best-score" (and so are "score," "moves," and "best-max-moves," see code example, below). Once a variable is introduced, it can take on and continuously change values. In this model, "best-score" is a variable that each of the players updates after each of its attempts to land the marble on the target line—it is a measure of their best score so far.

We will now look closer at examples of the NetLogo code. The purpose of the following section is not so much to teach the reader NetLogo as much as to convey that intuitive, yet not spelled out, ideas regarding phenomena under inquiry become explicit when one must articulate them in simple lines of code.

Diving into NetLogo code. Following is an example of the computer procedure "p-adjust." This procedure name is an abbreviation of "Piaget-adjust." It is a set of rules that instruct each virtual player how to operate under the "Piagetian" condition when that condition is activated from the model's interface and a marble has already settled. There are also special procedures for the Vygotskiian condition (v-adjust), the Piagetian-Vygotskiian condition (pv-adjust), and the random condition (r-adjust). (See, below, how the procedure begins with the infinitive form "to p-adjust" and ends with "end"; note that comments to the right of semicolons are ignored by the computer when it compiles, or "reads," the code).

```
to p-adjust
  ;; if your score is better, that's your new best, otherwise stick with the old
  if (score < best-score) [   ;; note that lower scores are better (closer to target line)
    set best-score score
    set best-max-moves max-moves
  ]
end
```

The agent "reading" this procedure does the following. It evaluates its *score* (how well it just performed in terms of its distance from the target line) and compares it to its *best-score* (the smallest distance it has personally achieved so far). If the *score* is less than the *best-score*, the player executes the following two actions that are enclosed between the brackets: (1) *set best-score score*: the player assigns to its *best-score* variable the value of the current *score*; and (2) *set best-max-moves max-moves*: the player assigns to its *best-max-moves*, which records how far the player threw its marble when it previously achieved its *best-score*, the value of *max-moves*, which records how far the player threw its marble on this current attempt. Note that in the case that the IF conditional it not satisfied, that is, when it is not the case that *score* < *best-score*, the player simply ignores the clause between the brackets. That means that the player does not change the values of its personal variables *best-move* and *best-max-moves*, retaining them for the following attempt, whereupon it will again make the same comparison and judgment.

What the *p-adjust* procedure essentially means is that the player is oblivious to all other players. It is only looking at its own achievement and possibly updating its "memory" accordingly. Under the Vygotskiian condition, however, the player looks around, and selects one of the other players in its vicinity. For the Vygotskiian and the Piagetian–Vygotskiian conditions, we implemented Vygotsky's construct of the *zone of proximal development* (ZPD). If a player's achievement on a given attempt is, say, 10 units away from the target, and the player sees a neighbor getting the marble only 3 units away from the target, the first player might imitate that neighbor's performance, as long as the ZPD is at least 7 units large. That is, if the ZPD is set at 5 units, then that neighbor's performance will be out of that player's ZPD (because 7 is larger than 5) and, so, the player cannot imitate it. In summary, if the selected neighbor player did better on this trial and this superior performance is within the player's ZPD, the first player adopts both the *score* and the *best-max-moves* of that neighbor. Otherwise, the player records its personal values from the current run. Under the Piagetian–Vygotskiian condition, the player

first checks its current *score* and if it's the best so far, the player updates its values. The player then

selects a neighbor and only takes on that player's values if the neighbor did better than itself and the

score is within the player's ZPD. Finally, under the Random condition, the player can only record its

own current values and only if they are completely on target.

Note that when the ZPD slider (see Figure 2, p. XX) is set at the value "0," a player cannot learn

at all from its neighbors, because any performance that is even just 1 unit superior to its own is beyond

the player's learning scope. Conversely, when the ZPD slider is set at "60" (the maximum possible

distance from the target line), the ZPD is not constituting any constraint on the prospects of imitating the

performance of neighbors. So, any values larger than "0" will increasingly enable the imitation of a

wider range of neighbor performance. That is, we expect that under the Vygotskiian condition,

individual learning, and, therefore, group learning, will increase as a function of the ZPD. The notion

that the ZPD is given to manipulation may be confusing unless the magnitude of the ZPD is interpreted

in proportion to the task: a small ZPD indicates that participants have much learning to do in order to

accomplish the task; a very large ZPD indicates that even the lowest-performing participants do not have

far to go.

[Insert Table 1 about here]

Table 1 summarizes the four conditions, displaying both the NetLogo procedure code and its

explanation.

Finally, we will focus on a line of code within the "to adjust" procedure (see Appendix A), in which players are "preparing" to throw the marble.[9] That is, they are each assigning to their personal variable, *max-moves,* a value that is based both on their prior learning, *best-max-moves*, and on a random element, which we have underlined (see below). There are variations on this code for the Piagetian and the Vygotskiian conditions, which reflect differences between these conditions, but the core code is as follows:

set max-moves (best-max-moves + <u>random-normal 0 (error * best-score / max-dist)</u>)

The underlined code primitive *random-normal* takes two arguments, a mean and a standard deviation, and reports a value based on those arguments. In the random-normal code clause, above, the mean is 0 and the standard deviation is a function of three variables: (1) *error*, which the user sets on the interface (see Figure 2, on p. 14); (2) *best-score*, which the player took from its last attempt (see in this section, above, the various conditions for computing this value); and (3) *max-dist*, which is a "global variable" (not personalized) and simply means the total horizontal length of the screen (how far a marble can go before it bounces back off the wall). Note that the standard-deviation argument takes two global variables, *error* and *max-dist*, and one player-specific variable, *best-score*. Recall that as the player improves, it achieves lower *best-score* values (the marble is stopping nearer to the target line). So the quotient of this player's *best-score / max-dist* is decreasing with the player's improvement. Thus, the standard-deviation, too, is decreasing. So what is this line of code modeling, and why did we choose to model this?

Each player has its own "personalized" distribution of anticipated performance that is based on its prior performance. As the player improves, it calibrates its performance, so the better the player, the higher the calibration. In other words, the error is normally distributed around the intended target, but

---

[9] For the purpose of this explanation, we have simplified the original code by collapsing two sub-conditions into a single condition and deleting an auxiliary procedure.

that distribution of error shrinks as the player approaches perfect performance. Think of a professional basket-ball player as compared to an amateur. From any given range, the professional will, on average, get the balls closer to the basket as compared to the amateur. But, also, the professional's errors *around* the basket will be smaller as compared to the amateur who will be throwing the ball "all over the place."

Finally, note that all the above explained the procedures within a single run. To produce the graphs in Figure 2 (see p. 14), the procedures were run over and over, with each procedure outputting values that are recorded in global lists (independent of the agents) and plotted onto the graphs and, periodically, onto the histogram (in the "Average Distance" plot, for example, each squiggly graph line records runs that are each of 30 group attempts). So the graph and histogram depict the aggregation of results from multiple runs under each of the four experimental conditions, making comparisons statistically meaningful.

Summary. We have now completed explaining the core procedures of the "I'm Game!" model, both in terms of the visible behaviors on the simulation interface and in terms of the code that underlies these rule-based behaviors. We have attempted, to the extent possible within the scope of this paper, to "glass-box" the model. That is, we have tried to present and explain the core procedures so as to afford the reader scrutiny of the model. The reader can now, hopefully, examine the code so as to judge its reliability (that it causes the agents to behave as we claim they behave) and its validity (that the agents' behavior reflects relevant key aspect of the real behaviors "out there," at least according to theoretical perspectives that we purport to be modeling).

We will now present results from running the experiment, discuss several emergent behaviors observed when operating the model, and end with concluding remarks.

Results

In earlier sections of this paper, we have delineated our motivation to build a computer-supported model of a socio-psychological phenomenon. Specifically, we discussed the advantage of

agent-based modeling (ABM) for studying complex phenomena, such as learning in situated contexts. Moreover, we argued that the process of building the model in and of itself constitutes an important factor of the learning that the model ultimately enables the researcher ("learning through modeling"). We then introduced the "I'm Game!" model of learning in situated contexts that we built in NetLogo and explained key elements of the programming code underlying the simulation. We now present results from running the model, including our insights from analyzing data from multiple runs under a variety of parameter settings.

Experimental Results

The "I'm Game!" model includes several parameters, such as the size of the execution error, the number of neighbors the player selects from, and the size of the ZPD. Therefore, the space of possible parameter-setting combinations for running this model is vast. Nevertheless, we will focus in this section on results from running the model under a specified range of settings, in each of which we gradually changed only one independent variable. Each specific parameter combination was run several times, and the values reported and displayed in graphs represent the collections of mean values from these iterations. Experimental results from running this simulation are as follows.

Main effect and control. The different strategies had main effects—they resulted in consistency within learning conditions and differences between them. The "Random" baseline condition consistently produced the weakest learning. Also, learning can occur under purely "Piagetian" or purely "Vygotskiian" conditions, but learning under these combined conditions exceeds learning gained under each strategy alone. Finally, whether the Piagetian learning is greater than the Vygotskiian learning or vice versa depends on combinations of the settings of the parameters "Vygotskiian-neighbors," "ZPD," and "error."

Error and learning. Under all three study conditions, we observe a strong direct relation between the execution error and learning (that is, an inverse relation between the error and the score; see Figure

3). This relation is stronger under the Piagetian condition than under the Vygotskiian condition. This Piagetian advantage suggests the utility of exploration when learners must rely on personal resources only.

[Insert Figure 3 about here]

When the error is low, Vygotskiian learners are liable to arrive at a group deadlock—all consistently undershooting or all consistently overshooting, as though they are converging on the target asymptotically yet cannot improve. Such deadlock is due to players' performance being bounded above by the performance of the best player in the group (and recall that the deviation is reciprocally dependent on the score, so a low error near the target line disallows radical change).

[Insert Figure 4 about here]

ZPD and learning. In the Vygotskiian mode, the larger the ZPD, the better the group average learning (see, in Figure 4, the diagonal "Vygotskiian" graph line representing the inverse relation between the ZPD value and the mean distance from the target). That is, the larger the ZPD, the more players could bootstrap their performance by imitating better performers in the group. Moreover, for a ZPD of 60, even the lowest-performing players had cognitive access to the performance of the highest-achieving players. This interpretation is supported by the observation that under extremely small ZPD settings, and especially when this is compounded with low "error" settings, some players are "left behind," because they initially perform poorly and then subsequently never find players within their narrow zone upon whom to model their performance. Note the consistent Piagetian performance.

[Insert Figure 5 about here]

Subgroup size and learning. In the Vygotskiian mode, and collapsed over the ZPD settings, the more neighbors each player observes, the faster the group learns (see Figure 5), because each player has a higher chance of imitating another player who is better than themselves. Note the interaction with the Piagetian condition that is designed so as not to be affected by the number of neighbors available for

imitation.

In summary, the "I'm Game!" model is reliable in that its purported effects, based on our programming of the agents' rule-based behaviors, bore out through running and analyzing the simulation. In particular, programming and simulating the *individual* agents' behaviors gave rise to *group* patterns of behavior that, in general, can be related to "real" behaviors, as seen through the perspectives of constructivist and socio-constructivist theoretical frameworks.

Conclusions and Limitations

We have attempted to demonstrate the viability of ABM for creating an intellectual space in which scholars from diverse disciplines can hone, share, and debate theories of psychology. To the extent that the simulation appears incomplete, perhaps these shortcomings can be articulated in terms of the simulation itself. For instance, if you believe that the model does not begin to capture the richness of Piaget's construct *schema* (accommodation, assimilation, specialization, equilibrium, etc.), or if you are concerned with the paucity of the marbles game to depict socially-mediated practices that a learner internalizes, according to Vygotsky's theory, we would welcome a critique that uses the existing model as a basis for expressing these constructs. Also, we encourage the readers to engage, themselves, in simulating cognitive-developmental/social phenomena from an agent-based perspective.

Through a case-study demonstration, we argued for ABM as a catalyst for inter-disciplinary, or, at the least, intra-disciplinary discourse. We set out by mentioning an anecdote of a previous study that was criticized, because it incorporated both Piagetian and Vygotskiian perspectives on learning. Our position was that when two or more perspectives can be *coherently* combined within a single study, such combination can only enrich the scope of the study. Rather than isolate these perspectives, it may be more rewarding to examine their potential complementarity, given the availability of powerful tools for bridging between these perspectives. To that extent, we have attempted to demonstrate that working in a programmable environment forces scholars from diverse backgrounds to couch their theories in common

terms. The modeling process highlights both what is shared by different theories and any critical differences between them. In particular, the "I'm Game!" simulation demonstrated how personal and interpersonal learning can be viewed and understood as reciprocal mechanisms. Both are at play.

The new lenses of ABM, we believe, will enable education researchers to explore, articulate, and share intuitions we have struggled to study rigorously and express coherently: the intuitions that individual intelligent behavior emerges from multi-componential cognitive dynamics and, at a "level up," that individuals and communities are interdependent through myriad dynamic reciprocities (e.g., Cole & Wertsch, 2002; Greeno, 1998, 2006; Fischer & Rose, 1999; Minsky, 1985).

The new lenses of ABM, we believe, will enable education researchers to explore, articulate, and share intuitions we have struggled to study rigorously and express coherently: the intuitions that individual intelligent behavior emerges from multi-componential cognitive dynamics and, at a "level up," that individuals and communities are interdependent through myriad dynamic reciprocities (e.g., Cole & Wertsch, 2002; Greeno, 1998, 2006; Fischer & Rose, 1999; Minsky, 1985).

Finally, we have focused on ABM primarily as a methodology for applying the complementary cognitive and social perspectives to research on psychology theory. Yet this methodology goes beyond a technique for coordinating data from different levels of observation—the methodology is embedded in and expressive of a fundamental tenet of complexity studies. Namely, ABM-based research methodology bespeaks the position that scientific and social phenomena are inherently complex—for each observational grain size there is a lower-level grain size of activity through which the observed phenomena emerge. Thus, even phenomena that do not appear fundamentally social such as a single individual's cognition can nevertheless be modeled as emerging from the activity of many lower-level cognitive elements (Blikstein, Abrahamson, & Wilensky, 2006; Blikstein, Wilensky & Hammer, 2006; see also Fischer & Rose, 1999; Minsky, 1985). From this perspective, agent-based models of human psychology are not definitive but recursive, like fractals that repeat their structure at lower and lower

orders of magnitude. Yet this endlessly collapsible and apparently volatile structure does not detract from the power of ABM. On the contrary: unlike many traditional models that are *static* at their level of description, agent-based models provide the tools for exploring microscopic–telescopic phenomenal structures, which *develop over time*, and for seeking rich explanatory coherence at multiple levels of observation.[10]

---

[10] "Led by a new paradigm, scientists adopt new instruments and look in new places…. It is rather as if the professional community had been suddenly transported to another planet where familiar objects are seen in a different light and are joined by unfamiliar ones as well" (Kuhn, 1962, p. 111).

References:

Axelrod, R. M. (1997). *The complexity of cooperation: Agent-based models of competition and collaboration*. Princeton: Princeton University Press.

Axelrod, R., & Cohen, M. D. (1999). *Harnessing complexity: Organization implications of a scientific frontier.* New York: Free Press.

Bar-Yam, Y. (1997). *Dynamics of complex systems*. Reading, MA: Addison-Wesley.

Blikstein, P., Abrahamson, D., & Wilensky, U. (2006, June). Minsky, mind, and models: Juxtaposing agent-based computer simulations and clinical-interview data as a methodology for investigating cognitive-developmental theory. Paper presented at the annual meeting of the Jean Piaget Society, Baltimore, MD.

Blikstein, P., & Wilensky, U., & Hammer, D. (2006). *Modeling manifold epistemological stances with agent-based computer simulation.* Manuscript under review.

Bonabeau, E., Dorigo, M., Théraulaz, G. (1999). *Swarm Intelligence: From natural to artificial systems*. London: Oxford University Press.

Cairns-Smith, A. G. (1996). *Evolving the mind: On the nature of matter and the origin of consciousness.* New York: Cambridge University Press.

Clark, A. (1997). *Being there: Putting brain, body, and world together again*. Cambridge, MA: MIT Press.

Cole, M., & Wertsch, J. V. (2002). *Beyond the individual–social antinomy in discussions of Piaget and Vygotsky*. Retrieved January 1 2002, from http://www.massey.ac.nz/~alock//virtual/colevyg.htm

Collier, N., & Sallach, D. (2001). *Repast*. University of Chicago (2001). http://repast.sourceforge.net .

Diermeier, D. (with Antonio Merlo). (2000). Government turnover in parliamentary democracies. *Journal of Economic Theory, 94*, 46-79.

Epstein, J., & Axtell, R. (1996). *Growing artificial societies*: MIT Press.

Fischer, K. W., & Rose, S. P. (1999). Rulers, models, and nonlinear dynamics. In G. Savelsbergh, H. van der Maas, & p. van Geert, (Eds.), *Nonlinear developmental processes* (pp. 197 - 212). Amsterdam: Royal Netherlands Academy of Arts and Sciences.

Fuson, K. C., & Abrahamson, D. (2005). Understanding ratio and proportion as an example of the Apprehending Zone and Conceptual-Phase problem-solving models. In J. Campbell (Ed.) *Handbook of mathematical cognition* (pp. 213 – 234). New York: Psychology Press.

Goldenberg, P., Lewis, P., & O'Keefe, J. (1992). Dynamic representation and the development of a process understanding of function. In G. Harel & E. Dubinsky (Eds.) *The concept of function:* Aspects of epistemology and pedagogy (pp. 235-260). Mathematical Association of America.

Greeno, J. G. (1998). The situativity of knowing, learning, and research. *American Psychologist, 53*(1), 5-26.

Greeno, J. (2005). Learning in activity. In R. K. Sawyer (Ed.), *The Cambridge Handbook of the Learning Sciences* (pp. 79 - 96). New York: Cambridge University Press.

Heidegger, M. (1962). *Being and time* (J. Macquarrie & E. Robinson, Trans.). New York: Harper & Row.

Hmelo-Silver, C. E. & Pfeffer, M. G. (2004). Comparing expert and novice understanding of a complex system from the perspective of structures, behaviors, and functions. *Cognitive Science*, 28, 127-138.

Holland, J. (1995). *Hidden order: How adaptation builds complexity*. Reading, MA: Helix Books/ Addison–Wesley.

Huizinga, J. (1955). *Homo ludens: A study of the play element in culture* (???, Trans.). Boston, MA: Beacon Press. (Original work published 1944 in German)

Hutchins, E. (1995). *Cognition in the wild*. Cambridge, MA: M.I.T. Press.

Jacobson, M. J. & Wilensky, U. (2006). Complex systems in education: Scientific and educational importance and research challenges for the learning sciences. *Journal of the Learning Sciences, 15*(1), 11-34.

Johnson, S. (2001). *Emergence: The connected lives of ants, brains, cities, and software*. New York: Scribner.

Kauffman, S. (1995). *At home in the universe: The search for the laws of self-organization and complexity.* Oxford: Oxford University Press.

Kuhn, T. (1962). *The structure of scientific revolutions*. Chicago: University of Chicago Press.

Langton, C. & Burkhardt, G. (1997). *Swarm*. Santa Fe Institute, Santa Fe. http://www.swarm.org/release.html

Lesh, R., & Doerr, H. M. (2003). *Beyond constructivism: Models and modeling perspectives on mathematics teaching, learning, and problem solving*. Mahwah, NJ: Lawrence Erlbaum Associates.

Levin, J., & Cole, M. (2006). *Simulations as mediators for distributed learning*. Manuscript in preparation.

Minsky, M. (1985). *The society of mind*. London, England: Hienemann.

Norman, D. A. (1991). Cognitive artifacts. In J. M. Carroll (Ed.), *Designing interaction: Psychology at the human-computer interface* (pp. 17-38). New York: Cambridge University Press.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York, Basic Books.

Piaget, J. (1962). *Play, dreams, and imitation in childhood* (C. Gattegno & F. M. Hodgson, Trans.). New York: Norton. (Original work published 1951)

Piaget, J. (1971). *Introduction to genetic epistemology*. New York: Norton.

Resnick, M. (1994). *Turtles, Termites and Traffic Jams: Explorations in Massively Parallel Microworlds*. Cambridge, MA: MIT Press.

Resnick, M. , & Wilensky, U. (1998). Diving into complexity: Developing probabilistic decentralized thinking through role-playing activities. *Journal of the Learning Sciences, 7*(2), 153-171.

Reynolds, C. W. (1987). *Flocks, herds, and schools: A distributed behavioral model, in computer graphics.* Paper presented at the 21(4) (SIGGRAPH '87 Conference Proceedings), pages 25-34.

Salk, J. (1983). Anatomy of reality: Merging of intuition and reason. In R. N. Anshen (Ed.), *Convergence*. New York: Columbia University Press.

Salomon, G., Perkins, D. N., & Globerson, T. (1991). Partners in cognition: Extending human intelligences with intelligent technologies. *Educational Researcher, 20*(3), , 2-9.

Schelling, T. C. (1971). Dynamic models of segregation. *Journal of Mathematical Sociology 1,* 143 – 186.

Solé , R. & Goodwin, B. (2000). *Signs of Life.* New York: Basic Books

Tisue, S., & Wilensky, U. (2004). NetLogo: Design and implementation of a multi-agent modeling environment. In *Proceedings of Agent 2004*, Chicago.
http://ccl.northwestern.edu/papers/agent2004.pdf

Troisi, A., Wong, V., and Ratner, M. (2005). An agent-based approach for modeling molecular self-

organization. *Proceedings of the National Academy of Sciences, 102*(2), 255–260.

Tudge, J. & Rogoff, B. (1989). Peer influences on cognitive development: Piagetian and Vygotskiian perspectives. In M. Bornstein & J. Bruner (Eds.), *Interaction in cognitive development* (pp. 17 – 40). Hillsdale, NJ: Erlbaum.

Vygotsky, L. S. (1962). *Thought and Language*. Cambridge MA: M.I.T. Press. (Original work, *Thinking and Speaking*, published 1934)

Vygotsky, L. S. (1978). In M. Cole, V. John-Steiner, S. Scribner, & E. Souberman (Eds.), *Mind in society: The development of higher psychological processes*. Cambridge: Harvard University Press. (Original work published 1935)

Wilensky, U. (1993). *Connected mathematics—Building concrete relationships with mathematical knowledge* Unpublished manuscript, Cambridge, MA: M.I.T.

Wilensky, U. (1997). What is normal anyway?: Therapy for epistemological anxiety. *Educational Studies in Mathematics, 33*(2), 171-202.

Wilensky, U. (1998). *NetLogo Flocking model*. http://ccl.northwestern.edu/netlogo/models/Flocking. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

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

Wilensky, U. (1999b). GasLab: An extensible modeling toolkit for exploring micro- and macro- views of gases. In N. Roberts, W. Feurzeig, & B. Hunter (Eds.), *Computer modeling and simulation in science education*. Berlin: Springer Verlag.

Wilensky, U. (2000). Modeling emergent phenomena with StarLogoT. *@CONCORD.org, (4) 1.*

Wilensky, U. (2001, April). *Emergent entities and emergent processes: Constructing emergence through multi-agent programming*. Paper presented at the annual meeting of the American Educational Research Association, Seattle, WA.

Wilensky, U. (2002). Modeling nature' s emergent patterns with multi-agent languages. Proceedings of *EuroLogo 2002*. Linz, Austria.

Wilensky, U., & Abrahamson, D. (2006, April). *Is a disease like a lottery?: Classroom networked technology that enables student reasoning about complexity.* Paper presented at the annual meeting of the American Educational Research Association, San Francisco, CA.

Wilensky, U., & Reisman, K. (1998). Learning biology through constructing and testing computational theories--An embodied modeling approach. Y. Bar-Yam, *Second International Conference on Complex Systems*, Nashua, NH, New England Complex Systems Institute.Wilensky, U., & Reisman, K. (2006). Thinking like a wolf, a sheep or a firefly: Learning biology through constructing and testing computational theories. *Cognition & Instruction,* 24(2), 171-209. http://ccl.northwestern.edu/papers/bio/long/.Wilensky, U., & Resnick, M. (1999). Thinking in levels: A dynamic systems approach to making sense of the world. *Journal of Science Education and Technology, 8*(1), 3-19.

Wilensky, U., & Stroup, W. (2003, April) *Embedded complementarity of object-based and aggregate reasoning in students developing understanding of dynamic systems*. Paper presented at the annual meeting of the American Educational Research Association, Chicago, IL.

Wilensky, U., & Papert, S. (2006). *Restructurations: Reformulations of knowledge disciplines through new representational forms*. (Manuscript in preparation)

Wolfram, S. (2002). *A new kind of science*. Champaign, IL: Wolfram Media, Inc.

Note: characters appearing to the right of two semicolons (";;") are comments on the code that the modeler wrote to help colleagues and fellow researcher make sense both of specific procedures and how they fit into the super-procedures. For explanations of the code language, see the NetLogo tutorial that comes with the free download of NetLogo at http://ccl.northwestern.edu/netlogo/. For an explanation of several key elements of this model's procedures, see the section *Computational Procedures That Implement a Theoretical Model.*

```
globals [ target   ;; the target line in the middle
       max-dist   ;; the maximum distance to the wall
       ticks-left   ;; the number of runs left of the current strategy
       pavg vavg pvavg ravg        ;; lists of each run's distance halfway through ticks-left
                          ;; ex. pavg = [ 2 3 17 ... ]
                          ;; this metric allows for comparison between strategies
    ]
turtles-own [ max-moves      ;; the max-moves of the current throw
         best-max-moves ;; the best throw of the agent (or, if "no memory," the current throw)
         moves-left    ;; the number of moves left until the ball comes to rest
         score        ;; the current score (how far from target line... lower is better)
         best-score   ;; best score (or, if no memory, current throw)
       ]

to setup
 ca
 set-default-shape turtles "circle-arrow"
 crt number-of-players
 ask turtles [ set color color + 0.1 ] ;; to make trails a little easier
 set pavg [0] set vavg [0] set pvavg [0] set ravg [0]
 ;; initialized with zeros so i can use mean from the start
 rerun
end

to rerun
 cp
 ;; random strategies lets you keep the simulation running picking from the available strategies
 ;;    at random
 if ( randomize-strategy-on-setup? ) [
    let new-strategy random 4
    if ( new-strategy = 3 ) [ set strategy "Random" ]
    if ( new-strategy = 2 ) [ set strategy "Piagetian" ]
    if ( new-strategy = 1 ) [ set strategy "Vygotskiian" ]
    if ( new-strategy = 0 ) [ set strategy "P-V" ]
 ]
 set ticks-left 30
 set max-dist ( screen-size-x )
 setup-plots
 setup-target
 setup-turtles
 display
end
```

```
to setup-plots
  set-current-plot "avg distance"
  set-plot-y-range 0 screen-edge-x  ;; most will fall within this range (all after a couple of steps)
  set-current-plot-pen strategy ppu plotxy -1 0 ppd

  set-current-plot "strategy avgs"
  set-current-plot-pen "Random" plot-pen-reset plotxy 0 precision mean ravg 2
  set-current-plot-pen "Piagetian" plot-pen-reset plotxy 1 precision mean pavg 2
  set-current-plot-pen "Vygotskiian" plot-pen-reset plotxy 2 precision mean vavg 2
  set-current-plot-pen "P-V" plot-pen-reset plotxy 3 precision mean pvavg 2
end

to-report get-strategy-color
  ifelse ( strategy = "Random" )
  [ report green ]
  [ ifelse ( strategy = "Piagetian" )
    [ report blue ]
    [ ifelse ( strategy = "Vygotskiian" )
      [ report red ]
      [ report grey ]
    ]
  ]
end

to setup-target
  ;; the target is the line in the center of the screen, 2 patches thick
  set target patches with [ abs ( pxcor ) < 1 ]
  ask target [ set pcolor get-strategy-color ]
end

to setup-turtles
  ask turtles
  [
    ;; spread out the turtles evenly
    set size ( screen-size-y / count turtles )
    setxy ( - screen-edge-x ) round ((who * screen-size-y / count turtles)) + 2


    set score 100000
    set best-score score
    set heading 90

    ;; their max-moves are randomly distributed over the length of the playing field
    set max-moves random max-dist
  ]
end
```

38

```
to go
  ;; the score is their distance from the target
  ask turtles [ set score evaluate-score ]  ;; the score is the distance from the target line
  ask turtles [ adjust ] ;; act according to the strategy, e.g,. in Vygotskiian, you compare your scores to a neighbor and
possibly update your score

  reset

  ;; move all the turtles forward to their max-moves spot.
  ;; we can't just say "fd max-moves" because we want them to bounce off the wall
  ;; + leave a dissipating trail
  ask turtles [ set moves-left limit-legal-distance max-moves ]
  let moving-turtles turtles with [ moves-left > 0 ]
  while [ any? moving-turtles ] [
    set moving-turtles turtles with [ moves-left > 0 ]
    ask moving-turtles [
      move-x
      if (trails?) [ stamp (color - 5) + ( 10 * ( max-moves - moves-left )  / max-moves ) ]
    ]
  ]

  do-plots

  if (ticks-left < 0) and limited-run? [ display stop ]
  if (ticks-left < 0) and not limited-run? [ display rerun ]
  set ticks-left ticks-left - 1
end


to move-x
  set moves-left moves-left - 1
  fd 1
  if ( pxcor >= (screen-edge-x - 1) )
    [ set heading 270 fd 2 ]
end

to-report evaluate-score
  report ( distancexy-nowrap 0 ycor )
end

to-report limit-legal-distance [ val ]
  report ( min ( list ( max-dist - 1 ) max ( list 0 val ) ) )
end

to do-plots
  set-current-plot "avg distance"
  let curr-mean mean values-from turtles [ evaluate-score ]
  plot curr-mean
  if (ticks-left = 15) [
    if strategy = "Piagetian" [ set pavg fput round curr-mean pavg ]
    if strategy = "Vygotskiian" [ set vavg fput round curr-mean vavg ]
    if strategy = "P-V" [ set pvavg fput round curr-mean pvavg ]
    if strategy = "Random" [ set ravg fput round curr-mean ravg ]
  ]
end
```

```
to reset
 cp
 setup-target
 ask turtles [ set heading 90 set xcor ( - screen-edge-x ) ]
 display
end

to adjust
  if strategy = "Random" [
     r-adjust
     set max-moves best-max-moves
     stop
   ]
  if strategy = "Piagetian" [ p-adjust ]
  if strategy = "Vygotskiian" [ v-adjust ]
  if strategy = "P-V" [ pv-adjust ]

  if strategy = "Vygotskiian"
  [
   set max-moves limit-legal-distance
      ( best-max-moves +  ( random-normal 0 ( error * best-score / max-dist ) ) )
   stop
  ]

  ifelse ( xcor > 0 ) [
    set max-moves limit-legal-distance
      ( best-max-moves +  (- abs random-normal 0 ( error * best-score / max-dist ) ) )
  ] [
    set max-moves limit-legal-distance
      ( best-max-moves + (abs random-normal 0 ( error * best-score / max-dist ) ) )
  ]
end

to p-adjust
 ;; if your score is better, that's your new best, otherwise stick with the old
 if (score < best-score) [
   set best-score score
   set best-max-moves max-moves
 ]
end

to v-adjust
 let fellow nobody
 while [ fellow = nobody or fellow = self ]
         [ set fellow turtle (who + ( - (#-Vygotskiian-neighbors / 2) ) + random (1 + #-Vygotskiian-neighbors ) ) ]
 ;; look randomly to one of your neighbors

 ;; if the score is better and it is within your ZPD, use their max-moves.
 ifelse (best-score > best-score-of fellow) and (best-score - ZPD <= best-score-of fellow)
 [
   set best-score best-score-of fellow
   set best-max-moves best-max-moves-of fellow
 ]
 [
   set best-score score
   set best-max-moves max-moves
 ]
end
```

```
to pv-adjust
  let fellow nobody
  while [ fellow = nobody or fellow = self ]  [ set fellow turtle (who + ( - (#-Vygotskiian-neighbors / 2) ) + random (1 + #-
Vygotskiian-neighbors ) ) ]
  ;; look randomly to one of your neighbors

  ;; maximize your own score and...
  if ( score < best-score )
  [
    set best-score score
    set best-max-moves max-moves
  ]
  ;; check it against your neighbor's score
  if (best-score > best-score-of fellow) and (best-score - ZPD <= best-score-of fellow)
  [
    set best-score best-score-of fellow
    set best-max-moves best-max-moves-of fellow
  ]
end

to r-adjust
  ;; random strategy changes max-moves to a random number x if it's not at the wall
  ;; where 0 < x < max-dist
  ;; if it is at the target, it stops changing.
  ifelse ( ( abs pxcor) > 0 )
  [
  ; ifelse (smart-random?)
  ;  [ set best-max-moves limit-legal-distance max-moves + random-normal 0 error ]
    ;[
    set best-max-moves ( random max-dist )
    ;]
  ] [
    set best-max-moves ( max-dist / 2 ) - 1
  ]
end
```

Author Note

Table 1.

NetLogo Agent Procedures for the Experimental Conditions and Explanations of These Procedures

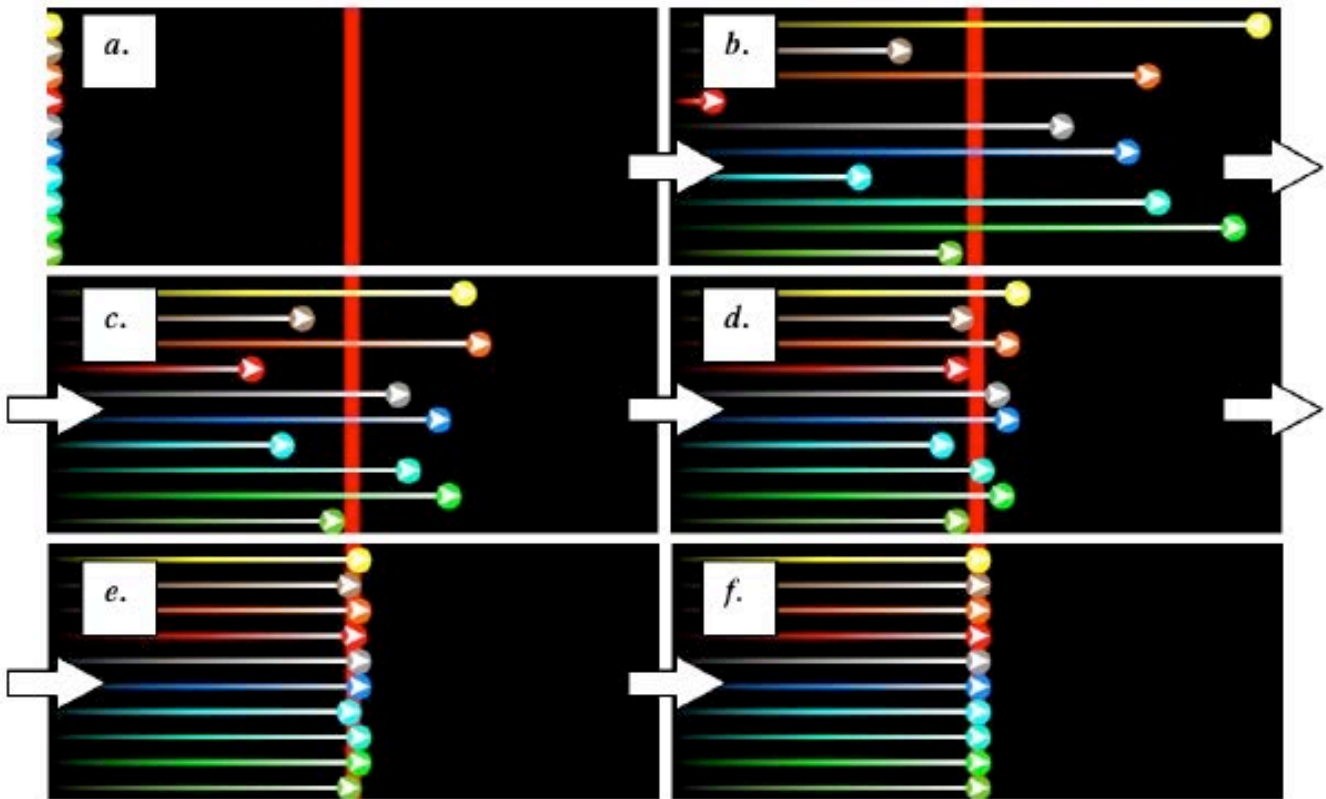| Procedure	Explanation

"Piagetian"
 	
``
to p-adjust
  if (score < best-score) [
    set best-score score
    set best-max-moves max-moves
  ]
end
``
	If you have performed better than in the past, record both how well you did now and how far you threw the marble.

"Vygotskiian"
 	
``
to v-adjust
  let fellow nobody
  while [(fellow = nobody) or (fellow = self)]
      [set fellow turtle (who + (- (#-Vygotskiian-neighbors / 2))
        + random (1 + #-Vygotskiian-neighbors)))
  ifelse (best-score-of fellow < best-score) and
      (best-score - best-score-of fellow <= ZPD)
  [
    set best-score best-score-of fellow
    set best-max-moves best-max-moves-of fellow
  ]
  [
    set best-score score
    set best-max-moves max-moves
  ]
end
``
	If a selected neighbor performed better than you and this advantage is within your ZPD, record both how well the neighbor did and how far the neighbor threw the marble. Otherwise, record your own values, regardless of how they compare to your own previous attempts.

"Piagetian–Vygotskiian"
 	
``
to pv-adjust
  let fellow nobody
  while (fellow = nobody) or (fellow = self)]
      [set fellow turtle (who + (- (#-Vygotskiian-neighbors / 2))
        + random (1 + #-Vygotskiian-neighbors))]
  if (score < best-score)
   [
    set best-score score
    set best-max-moves max-moves
   ]
  if (best-score > best-score-of fellow)
   [
    set best-score best-score-of fellow
    set best-max-moves best-max-moves-of fellow
   ]
end
``
	If you have performed better than in the past, record both how well you did now and how far you threw the marble. Then, if a selected neighbor performed better than you and this advantage is within your ZPD, record both how well the neighbor did and how far the neighbor threw the marble

"Random"
 	
``
to r-adjust
  if ((abs pxcor) > 0)
   [
    set best-max-moves (max-dist / 2) - 1
   ]
end
``
	If you have performed perfectly, record how far your threw the marble. Otherwise, don't record anything.

Figure 1. Snapshots from successive experimental runs of the NetLogo "I'm Game!" simulation of play-based "Piagetian" and/or "Vygotskiian" learning: Personal and interpersonal learning are reciprocal.
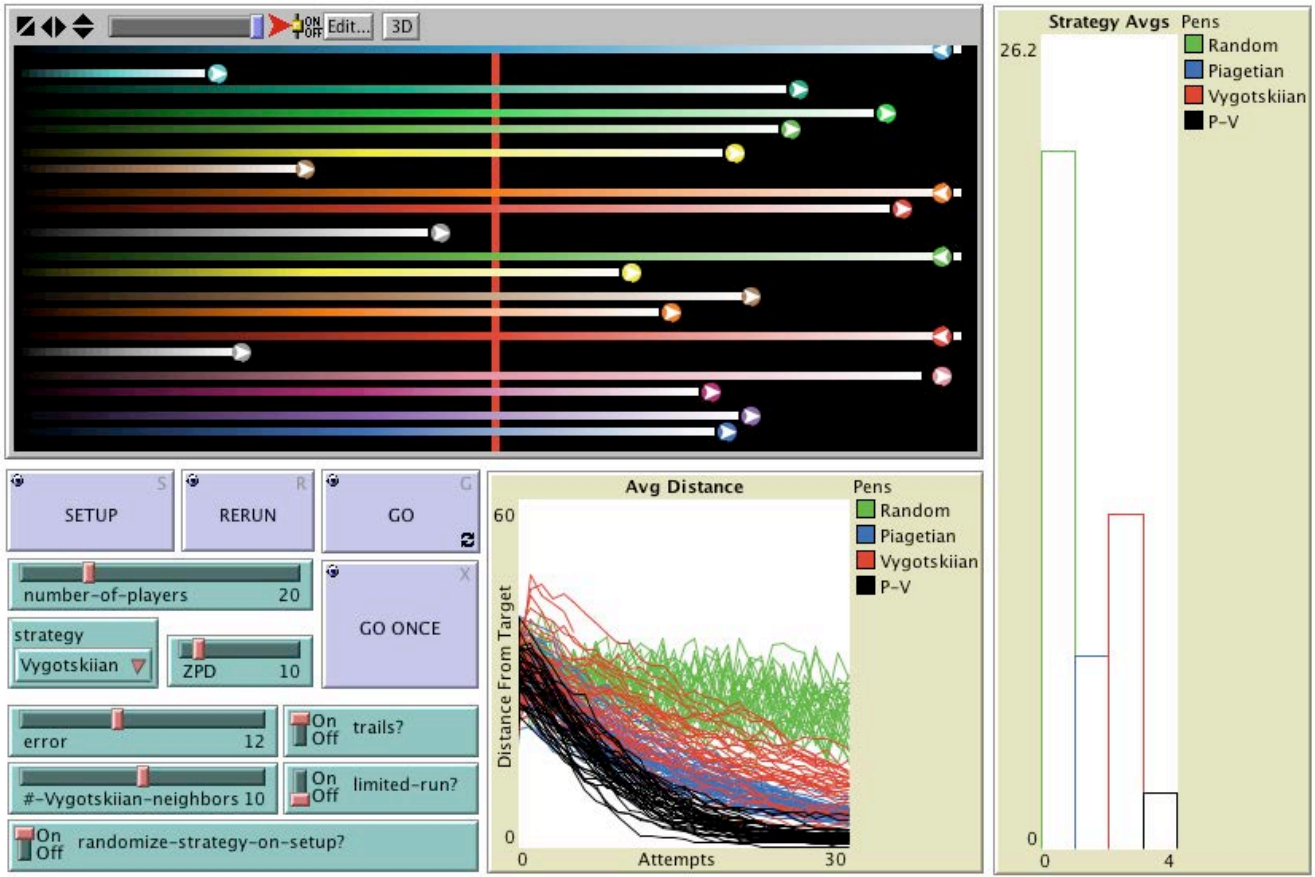
Figure 2. Interface of the NetLogo "I'm Game!" interactive simulation of learning. Under these particular experimental conditions, and aggregated over repeated runs, the group mean performance (distance from target) is ranked "Piagetian–Vygotskiian," "Piagetian," "Vygotskiian," and "Random."
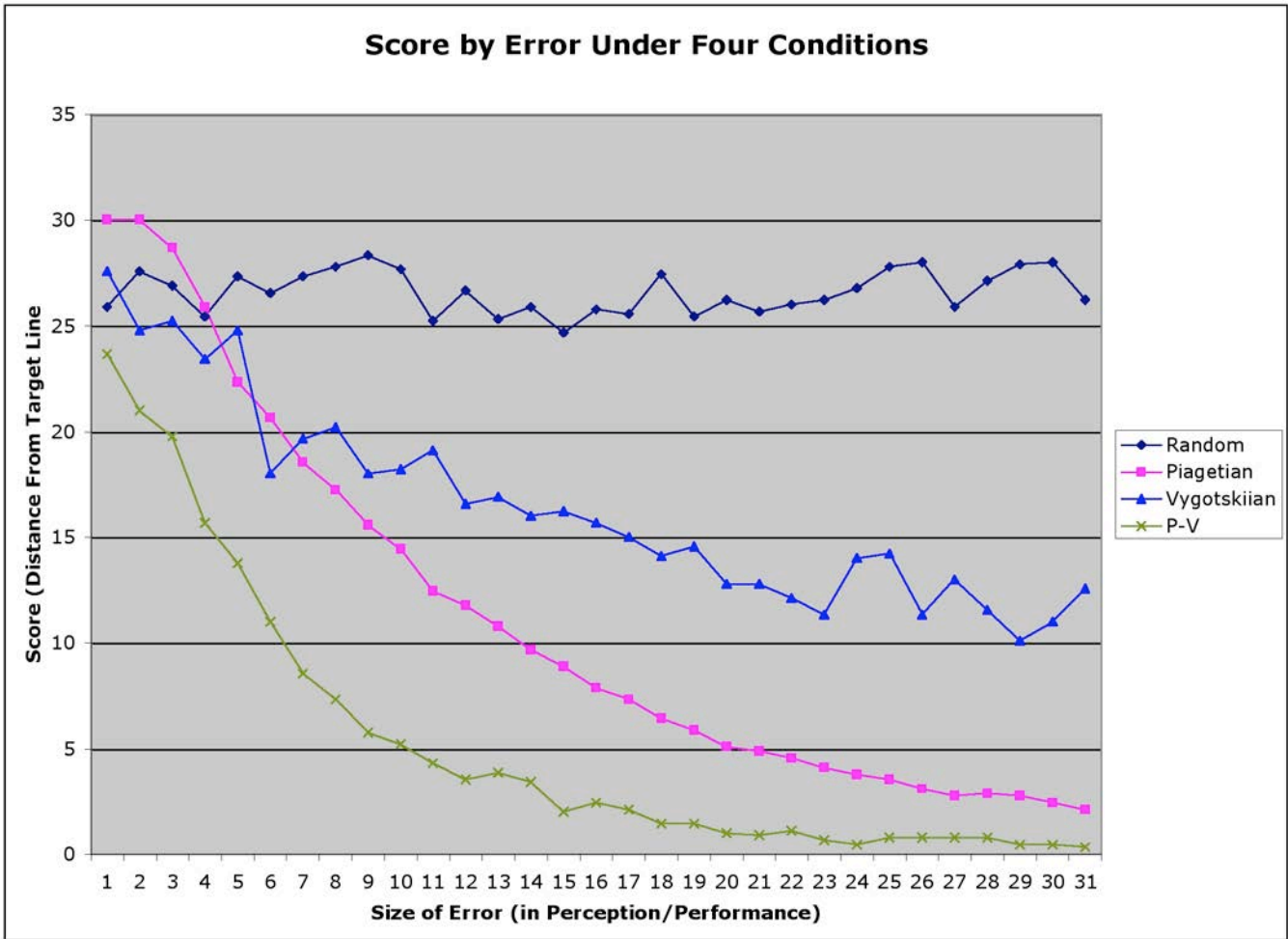
Figure 3. Increasing players' perception/execution "error" allows players to experience improved performance. The effect is more pronounced for the Piagetian (and hence the Piagetian–Vygotskiian) condition than for the Vygotskiian condition. In these experimental runs, there were 20 agents, the ZPD was set at 15, and the #-Vygotskiian-neighbors variable was at 4 (and 10 iterations were run per setting).
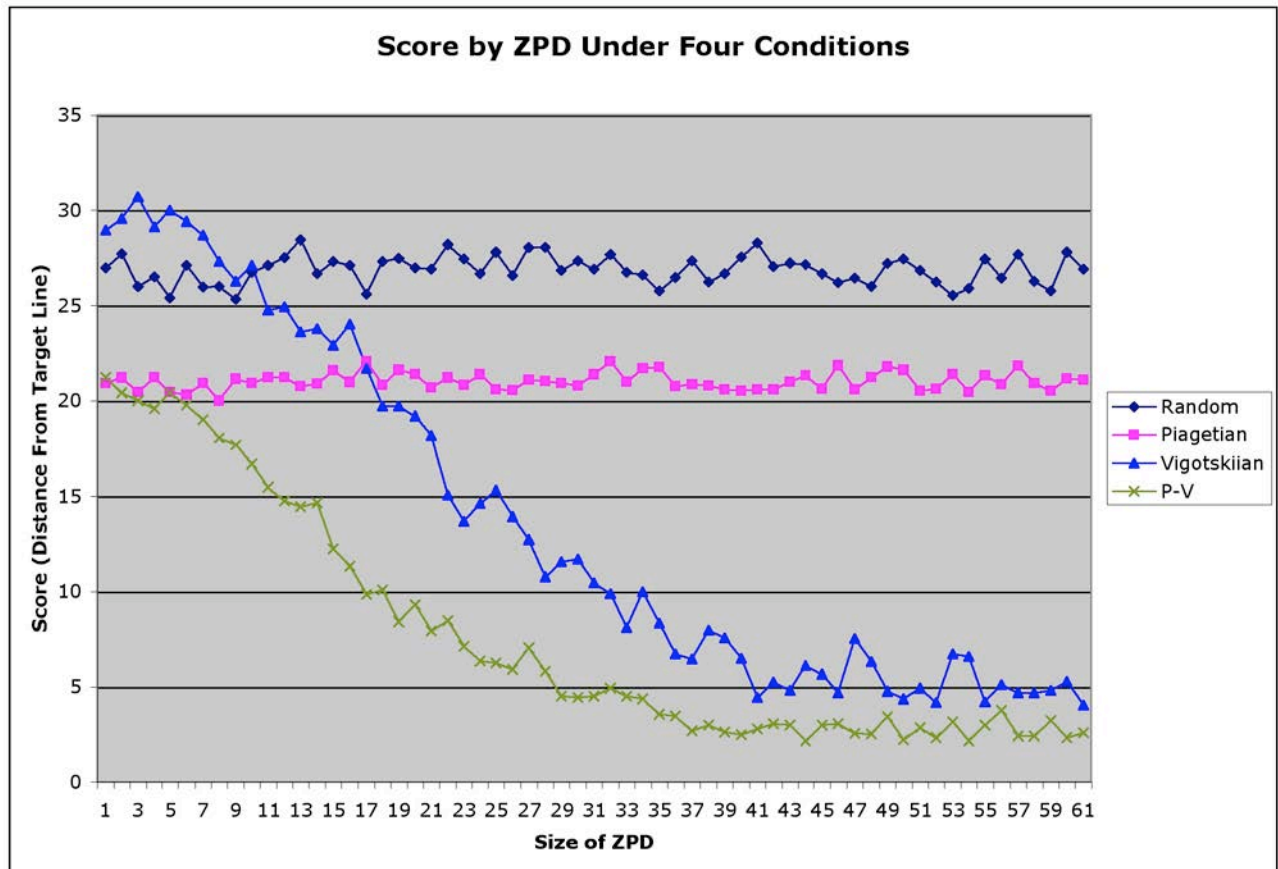
Figure 4. Broadening the ZPD setting increases learning opportunities and, thus, decreases the group mean score in the Vygotskiian and Piagetian–Vygotskiian experimental modes but does not affect the Random or Piagetian modes. In these experimental runs, there were 20 agents, the error was at 4, and '#-Vygotskiian-neighbors' was at 4 (and 30 iterations were run per setting).
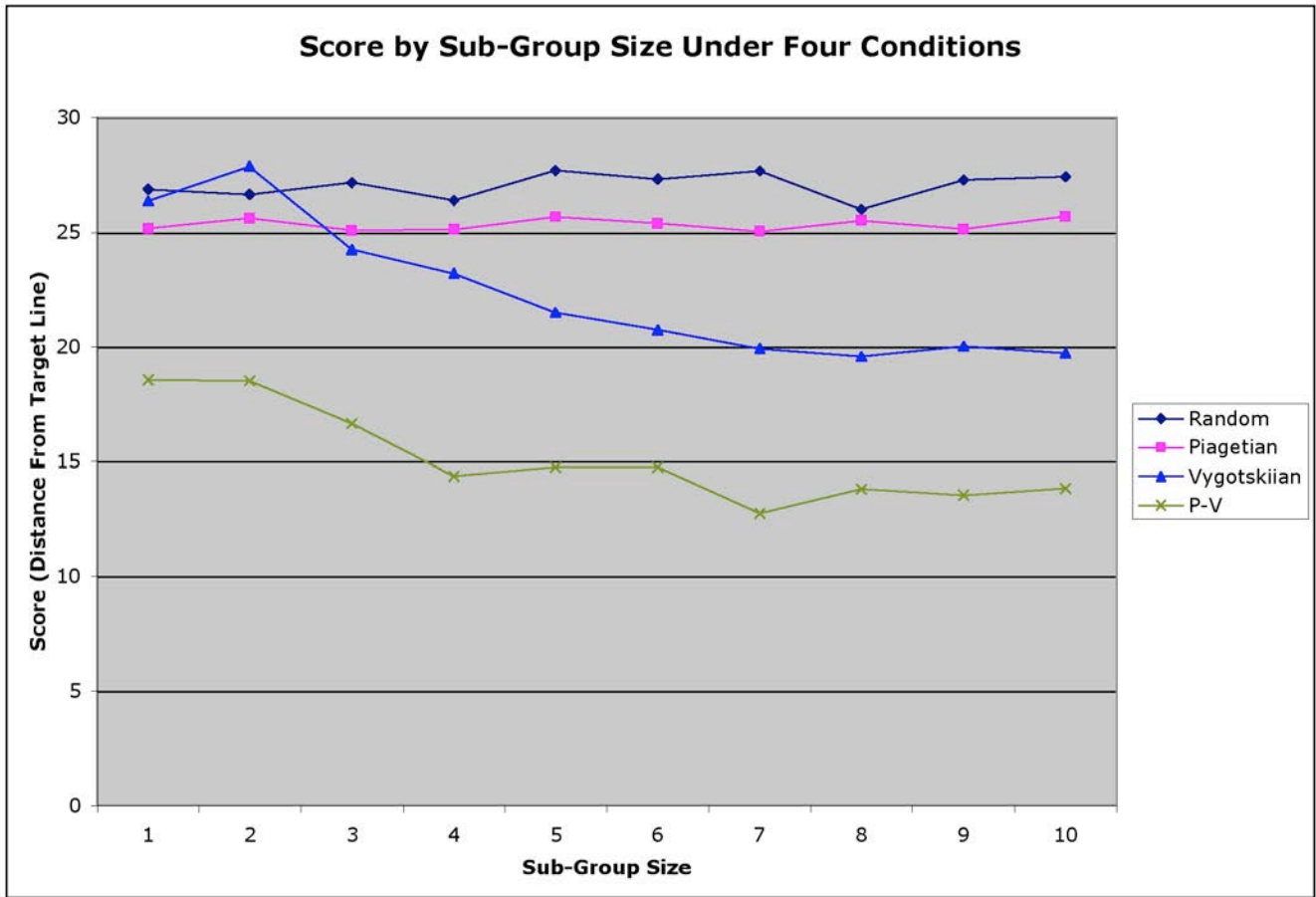
Figure 5. Increasing the number of each player's subgroup enhances individuals' chances to learn through imitation, under the Vygotskiian and Piagetian-Vygotskiian (but not Random or Piagetian) conditions. Therefore, the group mean score improves, too. In these experimental runs, there were 20 agents, the ZPD was set at 15, and the error variable was set at 3 (80 iterations per setting).