International Society of
the Learning Sciences

Quickstart Spaceship Programming for Developing Physical Intuition and Connecting it to Propositional Physics Knowledge

Jacob Kelter, Amanda Peel, Bradley Davey, Michael Horn, Uri Wilensky
jacobkelter@u.northwestern.edu, amanda.peel@northwestern.edu, bradleydavey2026@u.northwestern.edu, michael-horn@northwestern.edu, uri@northwestern.edu
Northwestern University, Center for Connected Learning and Computer Based Modeling

Abstract: Decades of research reveals that students enter physics classrooms with non-Newtonian intuitions that are difficult to change. We present a "quickstart" blocks-based programming environment in which students program the engines of a spaceship to navigate it to a new location. We then present a theoretical framework and analysis methodology combining student-produced computational artifacts with interaction analysis to investigate student learning in the environment. Preliminary results from a case study analysis of two students shows that through the programming task they align their intuitive mental models with Newtonian physics, and then by answering questions about the computational model afterwards, they were able to connect these intuitions to propositional physics knowledge.

Background

Decades of research shows that students enter the physics classroom with non-Newtonian intuitions and that changing these intuitions is difficult (e.g., Clement, 1982; diSessa, 1982; Halloun & Hestenes, 1985). To help students develop Newtonian intuitions we designed a spaceship programming microworld. A microworld embodies some domain of mathematics or science in a computational environment which students explore (Papert, 1980). A very early class of physics microworlds used the "Dynaturtle", a computational Newtonian object that students interact with by applying virtual "kicks" (diSessa, 1982; Papert, 1980). This early work showed that naive non-Newtonian physics intuitions are difficult to change but that playing Dynaturtle games can help. We build on this work by constructing a spaceship programming microworld in which students program the engines of a spaceship to turn on and off using programming blocks in a "quickstart programming environment" (Wagh & Wilensky, 2018). The programming blocks in a quickstart environment represent important concepts or mechanisms in a science domain rather than low-level computational primitives, enabling students to learn about scientific phenomena through programming without needing any prior programming experience. Two broad Constructionist learning principles embodied in the design are that (1) constructing an external "public entity" facilitates constructing mental knowledge structures (Papert, 1991) and (2) thinking and talking about one's activity aids learning (Harel & Papert, 1990). The microworld and domain-specific programming blocks are implemented with NetTango Web (Horn et al., 2020), a block-based interface to the agent-based programming environment NetLogo (Wilensky, 1999) that makes it easy for students to construct a public artifact, in this case, the program for the spaceship. The design also encouraged students to work in groups and discuss their problem solving.

In analyzing student learning we build on research focusing on the convergence of two collaborators' understanding (Roschelle, 1992). We combine this with the frameworks of mental models (Gentner & Stevens, 1983; Johnson-Laird, 2010) and representational redescription (Karmiloff-Smith, 1992; Taber, 2010) to analyze the convergence and divergence of two students' understanding. In our usage, a mental model (MM) is a dynamically constructed mental representation of a situation which can be "run" to make predictions. According to the representational redescription hypothesis (Karmiloff-Smith, 1992), children (and perhaps adults) develop increasingly explicit representations throughout development that enable increasingly flexible behavior. On this view, the representation a person initially develops to enable "behavioral mastery" in a domain is not replaced. Rather, representations are *redescribed* at more explicit levels. We assume that student MMs can contain multiple types of representations at different levels of explicitness.

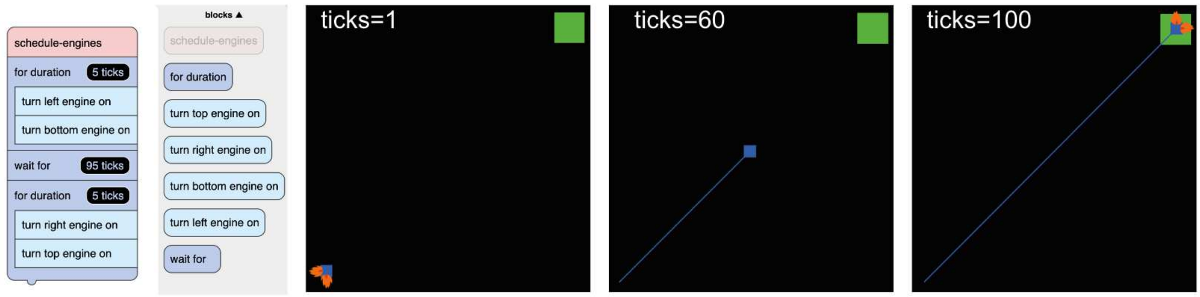
Following the Constructionist stance that a person's mental models are intimately linked with the external artifacts they construct, we analyze the alignments and misalignments between students' MMs and the computational model (CM) underlying the spaceship programming task. Using these frameworks, we aim to answer the following research question in future work: *How did students' MMs diverge and converge with each other and ultimately come to align with canonical physics knowledge embedded in the spaceship programming microworld?* The goal of our analysis method is to understand both what learning happened and how.

Methods

Software and task design

Students are given the task of programming the engines of a spaceship to reach a target square. They are encouraged to work in pairs to facilitate learning through discussion. In the microworld, the spaceship is visualized as a square with an engine on each side. Figure 1 shows the visualization along with a potential solution to get the spaceship to stop on the green square. The pane to the left shows the block-based programming environment. Learners drag blocks from the gray "blocks" area to program the pink "schedule-engines" procedure. In this case, the spaceship's bottom and left engine are scheduled to turn on for five "ticks" (the standard unit of time in NetLogo). Then the spaceship "waits" for 95 ticks without any engines on. Finally, the opposite engines are turned on for an equal amount of time to bring the spaceship to a stop. The blue line traces the path of the spaceship. This "diagonal" solution requires the fewest number of blocks, but many other solutions are possible. The next three panes in Figure 1 show the spaceship at the beginning, middle, and end of its journey with this block configuration.

Figure 1
A solution to the spaceship programming task.



Data collection

Data was collected in an AP physics A course in large public high school in the midwestern United States. Every block configuration (schedule-engines procedure) that each student tried was logged. Additionally, we recorded audiovisual data for the focal pair of students in this study, Kate and Jay, as they worked through the problem. Each student had their own laptop but they worked together.

Analysis methodology

We began qualitative analysis by transcribing seven minutes of audiovisual data of two students from a larger corpus. Next, a team of three researchers analyzed the video and transcript. We iteratively watched and segmented the video into 24 episodes of coherent joint and individual activity, each roughly 15-20 seconds in length. We summarized the result of each episode in terms of the students' problem-solving approaches and their understanding of the computational model and its latent physics concepts. Next, we deductively coded each episode for *convergence* or *divergence*. During this process we recognized and inductively coded emergent moments of *alignment* and *misalignment* between the two students' mental models (MM) and the computational model (CM). We present these deductive and inductive codes in Table 1. After coding the episodes, the first author wrote a narrative description of the episodes interleaving parts of the transcript (including students' computational blocks) with analysis. Other authors read this narrative description and any disagreements were discussed until agreement was reached. The combination of audiovisual data and a record of the students' computational blocks provides a rich combination for characterizing students' learning processes and trajectories.

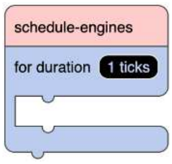
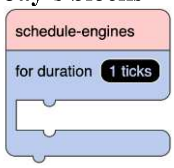
Table 1
Deductive (convergence) and inductive (divergence/alignment) coding scheme

Code	Description
MM convergence	The two students' MMs converge
MM divergence	The two students have different MMs
Blocks convergence	The two students set up the same programming blocks and parameters
Blocks divergence	The two students set up different programming blocks and/or parameters
Alignment	When a student's MM agrees with the CM
Misalignment	When a student's MM does not align with the CM

Example analysis

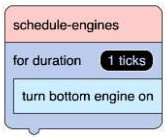
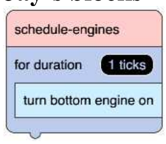
Thirty seconds after starting the activity, with the blocks in their default starting position, Kate realized they could move the ship diagonally. They *converged* on this with a brief exchange:

Figure 2
Initial convergence excerpt

	Kate's blocks	**Kate's MM**	**Jay's blocks**	**Jay's MM**
Kate (0:30): We only have to use certain engines (makes a triangle gesture with both hands together) **Jay** (0:35): Wait what do you mean? **Kate** (0:36): Go diagonally **Jay** (0:37): Ooh yes. Okay, I want to see how far **Kate** (0:42): How much does a tick go? **Jay** (0:43): Yeah. Okay, I'm gonna do just "bottom" (engine) for one tick.		Go diagonally❼ only certain engines		Go diagonally❼ only certain engines

It is important to note that Kate initially uses a triangular hand gesture and only then uses the words "go diagonally" to clarify. This indicates she first developed an explicit but non-verbal MM and only afterwards redescribed it in words. At this point Kate and Jay's MMs have *converged* on the idea to "go diagonally". Despite this, Jay decided to start by turning on just the bottom engine for one tick. After setting up the blocks and running the model, Jay saw that the ship did not slow down or stop after the engine turned off and concluded, "So, it looks like there's no friction. That's useful," and then they both laughed. Their shared laughter might suggest that Kate and Jay had converged on an understanding of "no friction," but when Kate ran the model herself for 30 seconds, it is clear there was actually a *divergence* between their mental models and a *misalignment* between Kate's MM and the CM:

Figure 3
Divergence and misalignment excerpt

	Kate's blocks	**Kate's M**	**Jay's blocks**	**Jay's MM**
Kate (1:52): Why does it keep going? Oh it's "for duration" // wait // Or, you have to stop it? It will just keep going? **Jay** (2:05): Yeah, yeah		Ship will keep going❼ unclear why		Ship will keep going❼ no friction

After running the model herself Kate got confused as to why the spaceship did not stop on its own, revealing the *misalignment* between her MM and the CM. She quickly realized that it will "just keep going" but phrased it as a question to Jay for confirmation. After his confirmation, we can infer that her MM was *aligned* with the CM in that the spaceship will not stop on its own, but it is not clear if she had connected this to the physical concept of "no friction" as Jay had. Based on her later reactions, she probably had not.

Preliminary results

Due to space restrictions, we are unable to present a full case study here. However, preliminary analysis shows Kate and Jay progressively *aligned* their mental models (MMs) with the computational model (CM) embedded in a physics microworld. At times their MMs and programming blocks *diverged* from one another, but ultimately, they *converged* on both a shared solution to the spaceship programming task and a shared MM of why the solution works. In answering subsequent questions in the assignment, they linked the behavior of the model to canonical physics concepts such as force, acceleration, and inertia, helping them connect the non-verbal understanding they gained from interacting with the microworld to more formal physics concepts, including previously disconnected propositional knowledge.

Discussion and future work

The spaceship programming activity helps students develop physics understanding, because there is immediate feedback if the spaceship does not move or stop as they expect given the sequence of programming blocks. This creates a feedback loop which allows the student to iteratively refine their MM of the situation to align with the physics embedded in the microworld. Working in pairs, students discuss their activity and can learn from each other both through spoken interactions and through sharing their computational artifacts. The block-based programming activity additionally gives students experience with a simple form of programming and computational thinking (CT), aligning with current science standards. Incorporating CT into physics also expands participation and advances equity in CT, because a much more diverse set of students take physics courses compared to dedicated courses on computer science (Wilensky et al., 2014).

The block-based programming activity advances a methodological contribution of relating students' MMs to their programming blocks. The full record of a student's block-based program gives insight into the student's learning trajectory, because the state of their program often reflects their MM. When paired with video data, we have shown it is possible to construct a narrative that accounts for the dynamic *alignments* and *misalignments* between students' MMs and the CM as well as *convergences* and *divergences* between the two collaborating students' MMs. These theoretical constructs and methods can be used to understand students' learning trajectories and patterns of collaboration. Future work will present full case studies of student learning using the theoretical frameworks and methods presented here.

References

Clement, J. (1982). Students' preconceptions in introductory mechanics. *American Journal of Physics*, *50*(1), 66–71. https://doi.org/10.1119/1.12989

diSessa, A. A. (1982). Unlearning Aristotelian physics: A study of knowledge-based learning. *Cognitive Science*, *6*(1), 37–75. https://doi.org/10.1016/S0364-0213(82)80005-0

Gentner, D., & Stevens, A. L. (Eds.). (1983). *Mental Models*. Psychology Press.

Halloun, I. A., & Hestenes, D. (1985). Common sense concepts about motion. *American Journal of Physics*, *53*(11), 1056–1065. https://doi.org/10.1119/1.14031

Harel, I., & Papert, S. (1990). Software Design as a Learning Environment. *Interactive Learning Environments*, *1*(1), 1–32. https://doi.org/10.1080/1049482900010102

Horn, M. S., Baker, J., & Wilensky, U. (2020). *NetTango Web*. Center for Connected Learning and Computer Based Modeling, Northwestern University. http://ccl.northwestern.edu/nettangoweb/

Johnson-Laird, P. (2010). Mental models and human reasoning. *Proceedings of the National Academy of Sciences*, *107*(43), 18243–18250. https://doi.org/10.1073/pnas.1012933107

Karmiloff-Smith, A. (1992). *Beyond modularity: A developmental perspective on cognitive science* (1995th ed.). MIT Press. http://gen.lib.rus.ec/book/index.php?md5=c6536e1190efba657621082a34722ddb

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas* (2nd ed). Basic Books.

Papert, S. (1991). Chapter 1: Situating Constructionism. In I. Harel & S. Papert (Eds.), *Constructionism: Research reports and essays, 1985-1990*. Ablex.

Roschelle, J. (1992). Learning by Collaborating: Convergent Conceptual Change. *Journal of the Learning Sciences*, *2*(3), 235–276. https://doi.org/10.1207/s15327809jls0203_1

Taber, K. S. (2010). Intuitions, Conceptions and Frameworks: Modelling Student Cognition in Science Learning. In M. S. Khine & I. M. Saleh (Eds.), *New Science of Learning: Cognition, Computers and Collaboration in Education* (pp. 163–182). Springer. https://doi.org/10.1007/978-1-4419-5716-0_8

Wagh, A., & Wilensky, U. (2018). EvoBuild: A Quickstart Toolkit for Programming Agent-Based Models of Evolutionary Processes. *Journal of Science Education and Technology*, *27*(2), 131–146. https://doi.org/10.1007/s10956-017-9713-1

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

Wilensky, U., Brady, C. E., & Horn, M. S. (2014). Fostering computational literacy in science classrooms. *Communications of the ACM*, *57*(8), 24–28. https://doi.org/10.1145/2633031