Computational Modeling in High School Physics First: Postcards from the Edge

Multiple initiatives contend that all students should master computational thinking, including the *Next Generation Science Standards*[1], the *K-12 Framework for Computational Thinking*[2], and Code.org[3]. In turn, many physics teachers have begun to explore a variety of approaches to integrating computational modeling through programming [4, 5, 6, 7, 8]. These activities go beyond graphical analysis tools and interactive simulations that have been a recent staple for physics educators.

In coordination with the American Modeling Teachers Association [9] and Bootstrap [10], the AAPT leads a project [11, 12] to explore the integration of computational modeling in Physics First [13] courses. The program, Computational Modeling in Physics First with Bootstrap (CMPF-B) has served 80 secondary-level physics teachers (Figure 1) through 3-week curriculum development and dissemination workshops. These teachers have learned to implement Modeling Instruction[14] in a way that leverages computational representations to encode each conceptual model. They have developed resources that support Modeling pedagogy (a method of teaching centered on the construction, validation and application of the fundamental models of physics) with Bootstrap (a program that uses computer science to teach algebra, data science, and now physics) (Table I). All of CMPF-B's student-facing curricular materials are available at https://www.compadre.org/precollege/CMP/.

Blending these initiatives has led to a rich professional development experience for teachers, and this marriage of Modeling and programming has helped students to learn physics by adding another type of representation; previously, students used graphical analysis, diagrams, verbal reasoning, and mathematical expressions to explore physics. With the integration of *Pyret* (accessible from Pyret.org), Bootstrap's language and user interface aligned with Modeling

Apple, L., Baunach, J., Connelly, G., Gahlhoff, S., Megowan Romanowicz, C., Vieyra, R., & Walker, L. (2021, 10). Computational modeling in the high school physics classrooms: Postcards from the edge. *The Physics Teacher, 59*(7), 535-539.

pedagogy, students learn to use another tool to develop their understanding of physics. Figure 2 illustrates an example of how computer programming is woven into multiple representations of the constant velocity model.

Our experience with teachers shows that the move to integrate computing is not only about teaching with a new tool. Integration leads to new insights about pedagogical practices and can shift the way teachers think about physics. However, we have also learned that integration takes a firm commitment to one or more personal goals for instructional change. We present a series of reflections from our participants to illustrate how they have achieved various goals for integration by showing what computational modeling looks like in their classrooms. Using these examples, we hope that other teachers might begin to explore the potential for programming in their own courses.

Glenda Connelly: Coordinating Representations

I was new to Modeling when I took the CMPF-B workshop. I had never before taken a coding class of any kind. Even after 38 years of teaching, watching my students use multiple approaches to represent the same phenomenon was a revelation. The combination of systems schemas, motion maps, line graphs, bar charts, lab activities, plus the integration of programming in *Pyret* empowered my students to develop conceptual models that represent their thinking.

The Constant Velocity Model unit beautifully develops the connections among displacement, time and velocity, and the integration of computing helps students grasp the relevance and usefulness of the algebraic method we use to find the slope of a linear fit to a scatter plot. In one of the first physics/programming activities, my students had to decide how to collect position-time data for battery-powered tumble buggies. They determined the buggy's

position at certain points in time and graphed their data to determine its velocity. Then, they used lab data to predict when and where their buggy would intersect with a buggy of different velocity from another group (Figure 3).

In *Pyret*, they created a simulation of two buggies just like the one from the lab, and used their code to determine when the buggies would meet. The function underlying the motion of the buggies is written as `next-x: ((v* delta-t) + x.` I convey this code to my students as a reimagining of the same relationship expressed in slope-intercept form, $y=mx+b$, but in which delta-t refers to the time interval from moment-to-moment rather than elapsed clock time.

To complete their set of constant velocity representations, students made motion maps to help them visualize the buggy's constant rate of change of position. *Pyret* coding reinforced concepts learned in lab, and provided them an opportunity to apply a mathematical representation. As a culminating project, the students wrote a `next-x` function to move a rocket horizontally across a moonscape at a constant velocity. This led to rich discussions about how self-driving cars and drones use programs like the `next-x` function to govern their motion.

The `next-x` function served as a platform for understanding ensuing models. Students extended their rocket simulation in the Uniform Acceleration Model unit. They were excited to move their rocket vertically by adding acceleration due to gravity. In the Balanced and Unbalanced Forces Model units, students added thrust to their rocket program to control direction and descent, enabling their rocket to land safely on the Moon.

In education, the goal is often to "cover the material," so that students are exposed to as much information as possible. Using computation forces us to slow down after all the planning, data gathering, and analysis (i.e., model construction). The last step, model application, is often

overlooked. Coding with *Pyret* gave my students the opportunity to achieve model application and make direct connections between science and math representations.

Lillian Apple: Motivation and Confidence

The CMPF-B workshop equipped me with new computational tools to bring into my classroom. Learning *Pyret* recalibrated my perception of physics instruction. Prior to this course, I had been a Modeler for my entire teaching physics career. I was no stranger to the discomfort that this approach often brought students initially, but it often led to a deeper understanding of the concepts. What I *was* unfamiliar with was programming—this placed me in a position of continuous uncertainty and inquiry throughout the workshop. I lacked a level of confidence I was accustomed to, and I noticed my moments of frustration. I often requested extra help, and I once found myself saying "I'm not there, yet," when it seemed many around me were. I had in fact begun to personify my very own students, and it was terrifying.

I try to put myself in this position every summer. I believe that it is often difficult for physics teachers to recall the struggle that students face learning this subject. The CMPF-B workshop not only provided me with a new pedagogical tool, a new group of colleagues from around the country, and an exceptional level of mentorship, it gave me a glimpse of a day in the life of my students.

As my school year began, I held tightly to this frame of reference. I allowed my students to see my vulnerability, and explained often that although I was not yet fluent in *Pyret*, or programming in general, this was not an obstacle to my own learning. I asked them to learn alongside me, emphasizing the helpfulness of receiving the interface's feedback on my code whenever I made a mistake. The results surprised me.

Apple, L., Baunach, J., Connelly, G., Gahlhoff, S., Megowan Romanowicz, C., Vieyra, R., & Walker, L. (2021, 10). Computational modeling in the high school physics classrooms: Postcards from the edge. *The Physics Teacher, 59*(7), 535-539.

Having taught in a private institution for nine years, my (perhaps biased) opinion, is that many students fear failure. They spend much time avoiding errors. One aspect of the Modeling pedagogy that I appreciate is how it helps students to identify their own errors, learning something more deeply from these through discourse and data analysis. My students watched me learn to navigate through coding alongside them and this gave them regular opportunities to teach *me*.

Because there was a learning curve to programming, we began by coding images of flags. Anyone wandering into the class might have been perplexed. What does coding the images of flags have to do with physics? As I learned, not only does coding flags familiarize the students with the design process of evaluating feedback to arrive at an efficient solution, it addresses concepts such as position, spatial reasoning, and problem solving, all of which are important in learning physics. It also allows students to work at various levels of programming skill while learning a new language. While some reached their comfort level with successful completion of Italy's flag, others were working on the European Union flag. Some students began to look for challenges immediately, and within two days I had received five versions of code for the American flag (Figure 4). I watched their confidence grow as I assigned coding simulations that paralleled physics concepts throughout the year. I soon realized that *Pyret* was helping my students think about problem-solving differently, and I admired how this computational aspect permeated their reasoning in other contexts.

Lucas Walker: Exploring the Limitations of Models

One important aspect of teaching scientific models that computational modeling has enabled me to address is helping students get a real sense of model boundaries. As a Modeler, my

class routinely involves discussions about *when a model applies.* However, I rarely presented my students with situations in which the model *didn't* apply. In light of *NGSS* Science and Engineering Practice #2: Developing and Using Models*,* which requires that students understand the role and limitations of models, I needed to do more than pay lip service to model boundaries.

I selected air resistance as a starting point. I had a two-phase progression in mind while designing the unit. First, I wanted to make sure that students were motivated to engage in computational modeling by a problem that was difficult to solve with algebra. Then, I wanted them to see the free fall model as a special case of the more complex *real-world* air drag model.

First, students analyzed the motion of falling coffee filters. Using a motion detector, we measured terminal velocities of stacks of coffee filters dropped from a few meters above the floor (Figure 5) and found a quadratic relationship between drag force and velocity. Then, I posed this question: From what heights must I drop stacks of different numbers of coffee filters, so that they hit the ground at the same time as a single filter falling 1 meter? This problem is algebraically intractable, creating the need for a computational model to represent how gravitational and drag forces combine to produce the motion of the filters. Students checked the validity of their programs by comparing graphs of the simulated data generated by *Pyret* to graphs of the real-world data collected by the motion detector to justify using their simulation to make predictions—and slow motion video showed that their predictions were (reasonably) accurate!

To come full circle on my original goal, (drawing a meaningful contrast between the air drag model and the simpler free fall model), we used *Pyret* to model the motion of falling objects. I introduced objects that didn't experience significant air resistance. Students found that mass did not produce differences in acceleration. Finally, they had to predict the motion of a falling object with an unknown drag coefficient (so they couldn't use their simulation to predict it). Solving this

problem, they argued vigorously together about whether this object behaved like one they had modeled previously. This argument was more authentic than any I have ever mediated on this subject.

As a coda for the unit, students read a newspaper article about computational modeling in climate science, which described the matching of multiple models to existing data and working collaboratively with other teams to make predictions or make sense of contrasts. They reflected on how this mirrored the class experience in modeling coffee filter motion. Many students said that they understood this article more deeply than they would have at the beginning of the school year, and some who had little affinity for coding acknowledged the usefulness of computational methods. This was the best application of computational modeling that I have yet deployed, and I will definitely use it again.

Sonia Gahlhoff: Gender Equity through Access and Engagement

Participating in this program gave me an opportunity to incorporate Modeling strategies into my physics teaching and expose my students—especially young women—to creating computational representations of motion. My high school offers no computer science courses, and only includes coding in a recently-added engineering program and an extracurricular robotics class. In incorporating computational modeling into physics, my goal was to improve equity and access to important computing skills for underrepresented groups, including young women, students of color, and non-native English speakers—more than half of my physics students.

Most of my students had never been exposed to coding in a classroom setting. Integrating computational modeling into mechanics helped them develop clearer pictures of motion, motivating and engaging traditionally underrepresented students in computing *and* physics.

Apple, L., Baunach, J., Connelly, G., Gahlhoff, S., Megowan Romanowicz, C., Vieyra, R., & Walker, L. (2021, 10). Computational modeling in the high school physics classrooms: Postcards from the edge. *The Physics Teacher, 59*(7), 535-539.

Collaboration is key to ensuring that all my students feel welcome. They work at gender-balanced or single-gender tables of 4-5 students. When coding, the table groups collaborate on whiteboards, then break into pairs to pair-program.

To examine computational modeling's impact on my students, I asked for written feedback. A few have said it enhanced their understanding of motion: "*Pyret* showed how objects can move at different rates including constant velocity, speeding up, and slowing down," and, "*Pyret* helped me understand the graphing model for motion by showing me how it would have an effect in real life." Others say they find coding helpful to their understanding of functions via inputs and outputs. One student shared that, "*Pyret* changed my understanding of motion by the specific numbers used as inputs." Some recognize that solving problems computationally helps them simplify problem-solving, especially in cases where math has been an obstacle: "... I can better understand how to deconstruct a math problem so it makes more sense after doing similar work on *Pyret*," and, "I was able to see how far an object was traveling without having to focus on doing a ton of math which made it easier to focus on the model rather than the work behind it."

Some students are surprised to like programming, and see it as potentially useful to them even if they do not pursue a science career: "Yes, I do [think it will be relevant to my future], because the technology is the future and coding is using it," and "...since I at least have a background knowledge of it, it can come in handy." Students were able to make connections with career interests as well: "...the major I choose is mechanical engineering. Most mechanical engineers have to deal with software driven machines, thus it is important to have some programming knowledge." I found that most students appreciated acquiring computing skills, and the collaborative atmosphere in our physics classroom allows them to learn in a low risk, engaging learning environment. Students know it is challenging, but this does not deter them!

Apple, L., Baunach, J., Connelly, G., Gahlhoff, S., Megowan Romanowicz, C., Vieyra, R., & Walker, L. (2021, 10). Computational modeling in the high school physics classrooms: Postcards from the edge. *The Physics Teacher, 59*(7), 535-539.

Through this experience, I have learned coding is an integral new tool enhancing students' understanding of motion. Computational modeling in my physics classes helps me expand equity of access to introductory coding. The experience has been important to my students' confidence and understanding. In our diverse student body, female students, especially, are now able to experience what was unavailable to them in our previous curriculum.

John Baunach: Enriching Content and Extending the Curriculum

Using Modeling is a difficult paradigm shift for many teachers, largely because it emphasizes depth of understanding over breadth of content. This choice of depth over curricular breadth remains one of its greatest hurdles for teachers interested in adopting this pedagogy. Even though I am mostly satisfied with my ninth grade course's focus on Newtonian mechanics, arguably the heart of physics, almost entirely, I still worry from time to time that I'm shortchanging my students by not showing them some of the more flashy and clever experiments. With such a heavy focus on simple mechanics and everyday objects, there is always a small voice in the back of my mind pleading with me to *do something more exciting!*, even if I must sacrifice quality instruction to do so.

This voice was nowhere to be found, however, as I watched my students create a computational model of Millikan's Nobel-winning oil drop experiment. The experiment is a subtle and elegant one: charged droplets of oil fall through an electric field created between two plates, the opposing forces on each drop balanced so that it has a constant velocity. Using just that information and the mass of the droplet, students can determine the electron's charge.

It is a difficult experiment to visualize for students, and practically impossible to recreate in a high school lab. Because it relies on understanding the electric field—a concept with which

even some physics teachers are uncomfortable—it's often relegated to a brief hand-wavy lecture in the middle of the E&M curriculum. Before this year, my introductory physics classes would have never seen this experiment; but now, with computational representation, they're recreating Millikan's *magnum opus* on the computer just after grasping Newton's First Law.

Beginning with starter code for the images, students use conditional statements to construct the electric field inside and around the plates. They tweak the strength of the electric field, much in the way Millikan did, until the drop falls at a constant velocity through the region between the plates (Figure 6). They then determine the electric force, the charge of the oil drop, and, finally, the charge of the electron.

As I moved from group to group, I was struck by the depth of the conversations occurring. In one group, two students debated whether or not an electric field would exist above and below the two plates, or just between them. In another group, three students argued about whose whiteboard had the correct free-body diagrams before, during, and after the drop's fall through the field. The third group finished building the simulation early, but did not trust their results—surely 10^{-19} C is far too small a number for an electron's charge, right? They wondered aloud what the smallest meaningful number in the universe might be.

Teaching students to create computational models is a curricular investment, but choosing curriculum is not a zero-sum game. Teachers worry about possibly losing content while building these computational skills; I would ask those teachers to think instead of the content that could be regained. What experiments can we save from the dustbin of scientific history? What concepts (like fields) can be introduced earlier with the right tools and representations? What questions might students ask, and what problems might they solve, if we give them more tools than we ever had ourselves?

Apple, L., Baunach, J., Connelly, G., Gahlhoff, S., Megowan Romanowicz, C., Vieyra, R., & Walker, L. (2021, 10). Computational modeling in the high school physics classrooms: Postcards from the edge. *The Physics Teacher, 59*(7), 535-539.

Conclusion

These stories from CMPF-B participants offer snapshots of the diverse ways bringing computational modeling into their classrooms has influenced the teaching and learning of physics. While our workshop experience has remained relatively coherent over the past few years, we find that teachers report very different kinds of affordances from this integration: allowing students new tools to express their thinking, enriching the physics concepts and process skills that can be taught, and achieving greater equity for students by introducing them to programming while displaying what it means to encounter a brand-new approach for learning science. We hope this glimpse into the world of integration in our own project can serve as an inspiration to others to try computational modeling in their own physics classes.

Acknowledgements:

The authors would like to acknowledge contributions to this work from the cohort of 80 middle and high school physics teachers who participated in the 2016, 2017, 2018, and 2019 summer workshops, Bootstrap team members Kathi Fisler, Shriram Khrishnamurthi, Benjamin S. Lerner,, Joe Gibbs Politz, and Emmanuel Schanzer, teacher developers Jess Dykes and Melissa Girmscheid, and to early partner Fernand Brunschwig of STEMteachersNYC for his support in acquiring funding for and hosting workshops in 2016, 2017, and 2018.

This material is based upon work supported by the National Science Foundation under Grant No. 1640791.

Apple, L., Baunach, J., Connelly, G., Gahlhoff, S., Megowan Romanowicz, C., Vieyra, R., & Walker, L. (2021, 10). Computational modeling in the high school physics classrooms: Postcards from the edge. *The Physics Teacher, 59*(7), 535-539.

References

1. Lead States. *Next generation science standards: For states, by states.* (The National Academies Press, Washington, DC, 2013).

2. K12cs.org. K-12 Computer Science Framework. https://k12cs.org/. (2016).

3. Code.org. State of Computer Science Education: Policy and Implementation. https://code.org/files/2018_state_of_cs.pdf. (2018).

4. AAPT. *Advancing Interdisciplinary Integration of Computational Thinking in Science: Conference Report.* https://www.aapt.org/Resources/upload/Computational_Thinking_Conference_Report_Final_200212.pdf. (American Association of Physics Teachers, College Park, MD, 2020).

5. M. D. Caballero, J. B. Burk, J. M. Aiken, B. D. Thoms, S.S. Douglas, E. M. Scanlon, & M. F. Schatz. "Integrating Numerical Computation into the Modeling Instruction Curriculum," *The Physics Teacher.* **52**(1), 38-42 (January 2014).

6. Odden, T. O. B. & Burk, J. Computational essays in the physics classroom," *The Physics Teacher,* **58** (4), 252-255 (2020).

7. Orban, C. M. & Teeling-Smith, R. M. "Computational thinking in introductory physics. *The Physics Teacher,* **58** (4), 247-251 (2020).

8. STEMCoding. *Welcome to the STEMCoding Project!* The Ohio State University. https://u.osu.edu/stemcoding/. (2020).

9. American Modeling Teachers Association. https://www.modelinginstruction.org/. (n.d).

10. Bootstrap. http://www.bootstrapworld.org (n.d.)

11. AAPT. Computational Modeling in Physics First with Bootstrap. https://aapt.org/K12/Computational-Modeling-in-Physics-First.cfm (2020).

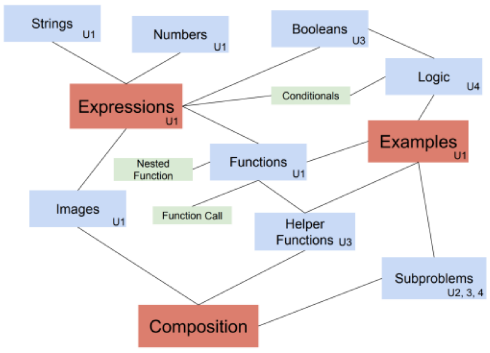
Apple, L., Baunach, J., Connelly, G., Gahlhoff, S., Megowan Romanowicz, C., Vieyra, R., & Walker, L. (2021, 10). Computational modeling in the high school physics classrooms: Postcards from the edge. *The Physics Teacher, 59*(7), 535-539.

12. Vieyra, R., Megowan Romanowicz, C., Fisler, K., Lerner, B. S., Gibbs Politz, J., & Krishnamurthi, S. (in preparation). Supporting Physics Teachers to Integrate Computational Modeling in their Instruction. *Journal of Science Teacher Education.*

13. AAPT. "AAPT Statement on Physics First." https://www.aapt.org/Resources/policy/physicsfirst.cfm. (2002).

14. M. Wells, D. Hestenes, D, & G. Swackhamer. "A modeling method for high school physics instruction," *The Physics Teacher,* **63** (7), 606-619 (1995).

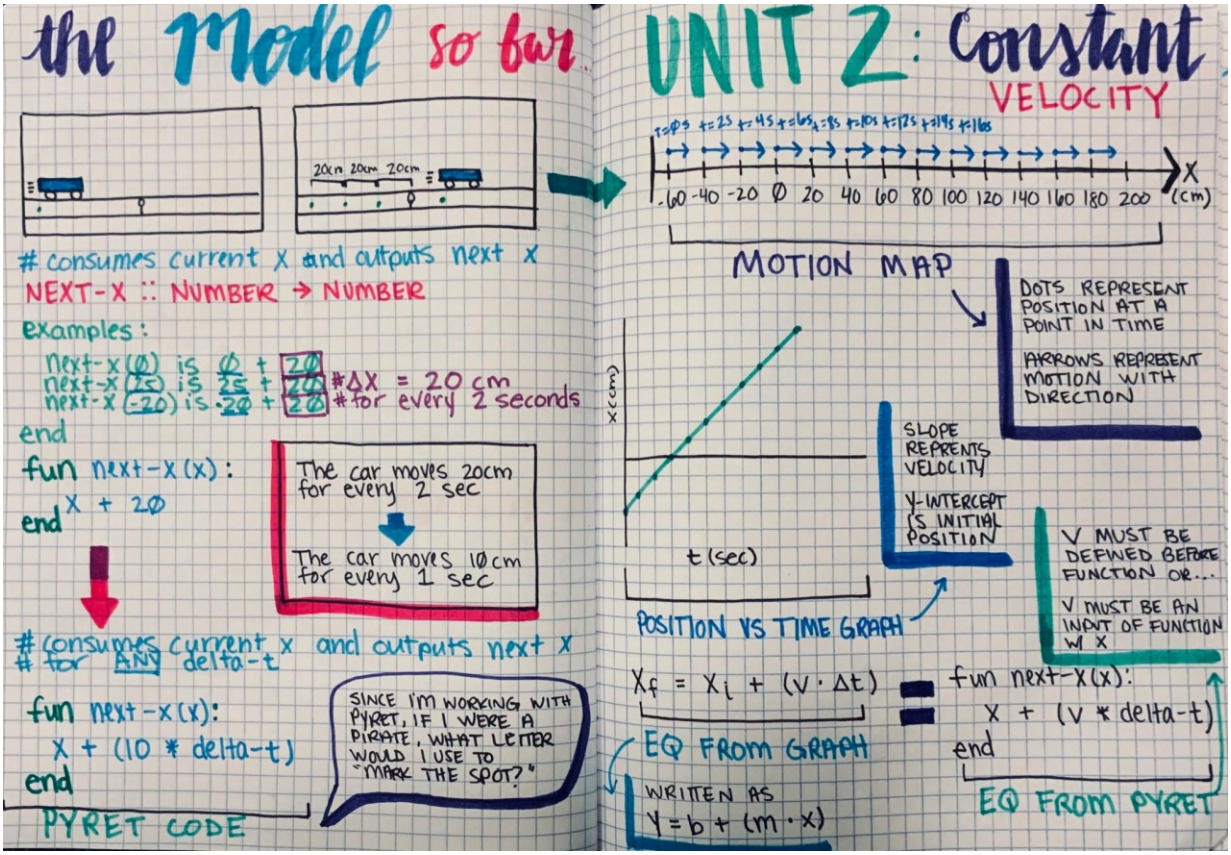
Table 1: Modeling Instruction and Bootstrap Descriptors

Modeling Instruction™	**Bootstrap**
Teachers use paradigm labs to help students construct conceptual models built around fundamental ideas in physics through multiple representations (data tables, graphs, algebraic equations, vector diagrams, pie charts) and consensus-based student discourse. Modeling Instruction typically offers 3-week workshops led by experienced high school Modelers.	Teachers help students to construct a variety of models by building code, often with the end goal of modeling motion and forces. Student learning is scaffolded through a "Design Recipe" as they attempt to solve word problems through mathematical modeling with *Pyret* [15], a language whose syntax and function has been designed to reinforce algebraic understanding. Bootstrap offers 3-day workshops, traditionally for math teachers who want to integrate programming into their curriculum.
Modeling Instruction Physics First Topics	*Bootstrap Computational Topics*
• Qualitative Energy Model (U1) • Constant Velocity Model (U2) • Uniform Acceleration Model (U3) • Balanced Forces Model (U4) • Unbalanced Forces Model (U5)	

Apple, L., Baunach, J., Connelly, G., Gahlhoff, S., Megowan Romanowicz, C., Vieyra, R., & Walker, L. (2021, 10). Computational modeling in the high school physics classrooms: Postcards from the edge. *The Physics Teacher, 59*(7), 535-539.

Figure 1. CMPF-B 2019-2020 cohort.

Figure 2. Illustration of multiple representations used in the CMPF-B Constant Velocity Model unit. (Credit: Lauren Stewart).



Apple, L., Baunach, J., Connelly, G., Gahlhoff, S., Megowan Romanowicz, C., Vieyra, R., & Walker, L. (2021, 10). Computational modeling in the high school physics classrooms: Postcards from the edge. *The Physics Teacher, 59*(7), 535-539.

Figure 3. Students use a *Pyret* program to predict the location of the collision of two buggies.

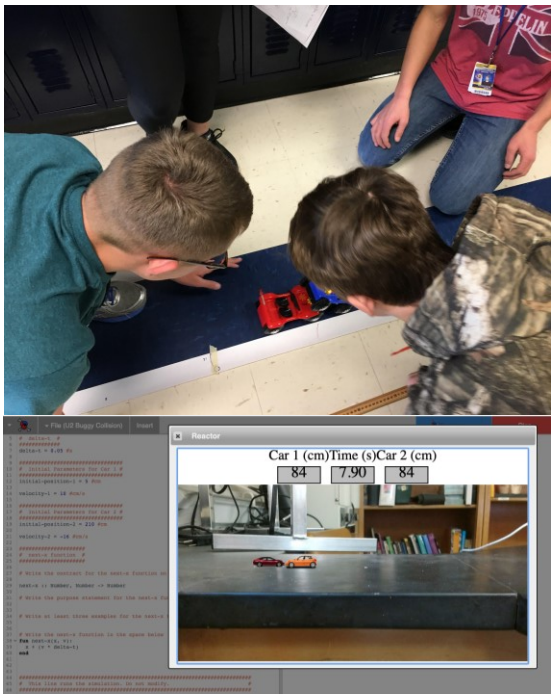
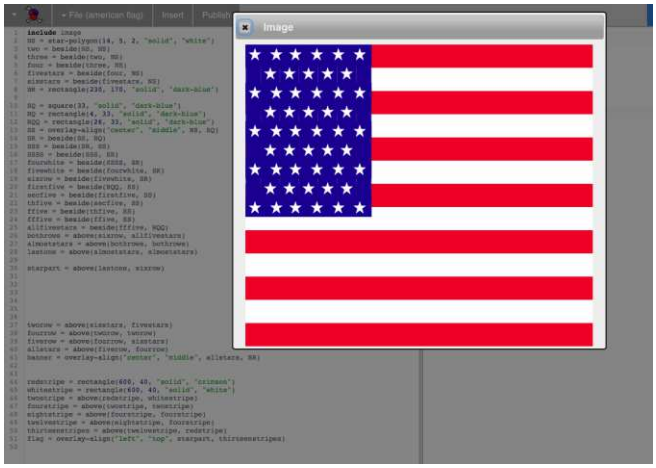


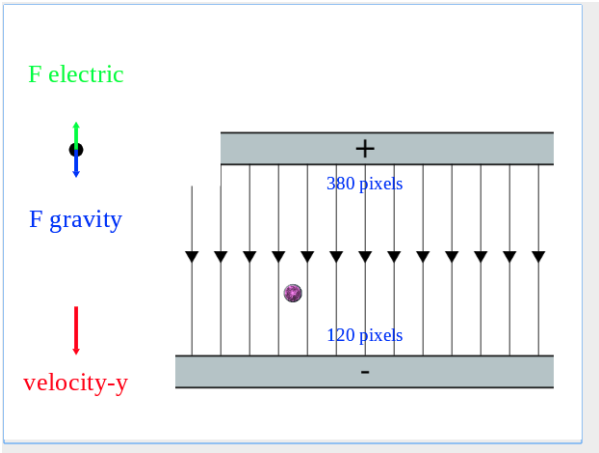
Figure 4. *Pyret* program for the design of the American flag.



Apple, L., Baunach, J., Connelly, G., Gahlhoff, S., Megowan Romanowicz, C., Vieyra, R., & Walker, L. (2021, 10). Computational modeling in the high school physics classrooms: Postcards from the edge. *The Physics Teacher, 59*(7), 535-539.

Figure 5. Coffee filter drop.

Figure 6. Screen capture from a *Pyret* simulation of the Millikan oil drop experiment.



Apple, L., Baunach, J., Connelly, G., Gahlhoff, S., Megowan Romanowicz, C., Vieyra, R., & Walker, L. (2021, 10). Computational modeling in the high school physics classrooms: Postcards from the edge. *The Physics Teacher, 59*(7), 535-539.