

Scaffolded Training Environment for Physics Programming (STEPP)

Modeling High School Physics using Concept Maps and State Machines

Midori Kitagawa

School of Arts, Technology and Emerging Communication
University of Texas at Dallas
Richardson, Texas
midori@utdallas.edu

ABSTRACT

Paul Fishwick

School of Arts, Technology and Emerging Communication
University of Texas at Dallas
Richardson, Texas
paul.fishwick@utdallas.edu

Michael Kesden

Department of Physics
University of Texas at Dallas
Richardson, Texas
kesden@utdallas.edu

Mary Urquhart

Department of
Science/Mathematics Education
University of Texas at Dallas
Richardson, Texas
urquhart@utdallas.edu

Rosanna Guadagno

Center for International Security Cooperation
University of Texas at Dallas
Richardson, Texas
Stanford, California
rong.jin@utdallas.edu
rosannaguadagno@stanford.edu

Rong Jin

Department of Computer Science and
University of Texas at Dallas
Stanford University
rong.jin@utdallas.edu

Erik Omogbehin

Department of Computer Science
University of Texas at Dallas
Richardson, Texas
erik.omogbehin@utdallas.edu

Aditya Prakash

Department of Computer Science
University of Texas at Dallas
Richardson, Texas
aditya.prakash2@utdallas.edu

Baily Hale

Department of
Science/Mathematics Education
University of Texas at Dallas
Richardson, Texas
bgh140130@utdallas.edu

Ken Suura

Department of
Science/Mathematics Education
University of Texas at Dallas
Richardson, Texas
kds140430@utdallas.edu

James Stanfield

Richardson High School
Richardson, Texas
jcstanfield@risd.org
henry.vo@risd.org

Henry Vo

Richardson High School

Aniket Raj

Game Design & Simulation
Lone Star College
Houston, Texas
research@aniketraj.com

We are a year into the development of a software tool for modeling and simulation (M&S) of 1D and 2D kinematics consistent with Newton's laws of motion. Our goal has been to introduce modeling and computational thinking into learning high-school physics.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise,

There are two main contributions from an M&S perspective: (1)

or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org. SIGSIM-PADS'19, June 3–5, 2019, Chicago, IL, USA.

© 2019 Association of Computing Machinery.

ACM ISBN 978-1-4503-6723-3/19/06...\$15.00.

<https://doi.org/10.1145/3316480.3325513>

the use of conceptual modeling, and (2) the application of Finite State Machines (FSMs) to model physical behavior. Both of these techniques have been used by the M&S community to model highlevel "soft systems" and discrete events. However, they have not been used to teach physics and represent ways in which M&S can improve physics education. We introduce the NSF-sponsored

STEPP project along with its hypothesis and goals. We also describe the development of the three STEPP modules, the server architecture, the assessment plan, and the expected outcomes.

CCS CONCEPTS

- Social and professional topics → Computing education
- Social and professional topics → K12 education
- Social and professional topics → Computational Thinking
- Human-centered computing → Human Computer Interaction (HCI)

KEYWORDS

Scaffolding, Physics, Programming, Computational Thinking, Modeling, Simulation, Concept Maps, Finite State Machines, High School, STEM.

ACM Reference format:

Midori Kitagawa, Paul Fishwick, Michael Kesden, Mary Urquhart, Rosanna Guadagno, Rong Jin, Ngoc Tran, Priyanka Awaraddi, Aditya Prakash, Baily Hale, Ken Suura, Aniket Raj, Erik Omogbehin, James Stanfield and Henry Vo, 2019. Scaffolded Training Environment for Physics Programming (STEPP): Modeling High School Physics using Concept Maps and State Machines. In *Proceedings of ACM SIGSIM PADS'19. ACM, Chicago, IL, USA, June 2019 (SIGSIM PADS'19)*, 10 pages. <https://doi.org/10.1145/3316480.3325513>

1 Introduction: STEPP Project

The Next Generation Science Standards (NGSS) [1] identify "Developing and using models" and "Use of mathematics and computational thinking (CT)" as two core practices of science and engineering. Although educators, researchers, and policy makers widely recognize the importance of modeling and CT or "the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent [2]," the introduction of these concepts into K-12 STEM education is still in an early stage. With funding from the NSF STEM + Computing Program, an interdisciplinary team of faculty and student researchers from Arts & Technology, Computer Science, Physics, Science/Math Education, and Psychology, along with high school physics and computer science teachers are collaborating on development of our Scaffolded Training Environment for Physics Programming (STEPP). STEPP will help students learn physics and cultivate CT skills by allowing them to create their own simulations using Concept Maps and Finite State Machines.

Scaffolding is an instructional method with two key aspects: (a) providing structure and support for completing the task, and (b) gradually removing supports so that the student can independently solve the problem [3]. Concept maps (CM) are diagrams that illustrate relationships among concepts have long been used in K-12 education [4,5,6,7]. Finite state machines (FSMs) are a restricted class of CMs that have been used to design

algorithms and teach programming and engineering [8,9,10]. CMs and FSMs are effective in teaching CT because they help students to learn integral elements of CT including abstractions, structured problem decompositions, iterative thinking, conditional logic, and efficiency. Scaffolding and programming with FSMs will allow students to focus on aspects of programming that complement the physics learning process. The Unity game engine will serve as the development platform for STEPP learning modules because of the low threshold of prior programming experience, high ceiling for learning potentials, and transferability to other applications.

Our hypothesis is that by constructing their own FSM modelbased simulations, students learning in a scaffolded synergistic environment will master physics concepts and CT more successfully than students learning with pre-made simulations. This hypothesis is one related to model editing as a value-added component rather than only model execution or simulation. If this hypothesis is not rejected in the experiment, then this will make a key contribution to the area of education within the modeling and simulation field. The modeling and simulation field has little empirical evidence of the effects of modeling and model design on humans. STEPP may assist in physics education, but it also is intended to validate the use of model design in a specific domain (i.e. physics). We are designing and implementing three STEPP modules with learning content aligned with Texas standards and NGSS (detailed in Sec. 5 STEPP Modules). The modules and supplemental information will be distributed by a server (detailed in Sec. 4 Architecture). A summer institute will be held for inservice and pre-service teachers to learn and experience STEPP and incorporate it into their own curricula and assessments. The STEPP modules will be tested at local high schools (detailed in Sec. 6 Assessment Plan). The impact that STEPP may have on modeling and simulation will be examined in the next section.

2 STEPP's Impact on Modeling and Simulation

There are numerous simulation programs for understanding physics. One early platform is based on Physlets which is now in its 3rd edition [11]. Another major effort is the PhET interactive simulations for physics [12]. Both Physlets and PhET are impressive in terms of scope and ease of use. A main benefit is to supplement classroom instruction with simulations that are interactive in nature. A student can learn about velocity, acceleration, and forces through virtual experimentation on a computer.

The field of modeling within the practice of simulation [13,14] has a fit within the sphere of interactive simulations. The simulations use models of physical behavior, in the form of pictures and animations. There are no formal modeling techniques in Physlets or PhET other than equations or equational terms. Equations and mathematical notation are the lingua franca of natural science and physics, so the use of this notation for modeling is standard practice. And yet, can we improve upon the modeling practice through insights gained from decades of M&S and the systems approaches therein?

We oriented our original STEPP grant proposal toward visual programming through the use of FSMs. The parlance of visual programming is identical to system modeling where the system model is diagrammed. The state machine comes from computer science and is found in academic subjects such as digital design (sequential machines), software engineering (e.g. state diagram in the Unified Modeling Language), and automata theory within the Chomsky hierarchy of machines and languages.

The main hypothesis behind our grant proposal was that the state machine could be a useful, discrete, method for describing and learning about physical behavior. Since classical physical quantities vary with time, the FSM was a novel introduction to high school physics: learning physics by way of a sequence of states separated by event transitions. Since the inception of the project, the research team considered other types of models. For instance, object-oriented design models were originally thought to be a useful way for the student to organize knowledge about the physical scenario. While we have yet to employ this type of design, we have embarked on a scaffolded modeling approach based on conceptual modeling [14,15]. The students begin with natural language constructs about the kinematic system. They gradually turn the natural language (e.g., moving left, speeding up) into physics nomenclature (e.g., velocity, acceleration) and then construct a linear state diagram where the states denote a duration of time and the transitional events are based on a Boolean condition involving time, displacement, or velocity.

3 STEPP's requirements

To implement STEPP, we first needed to recognize all the software requirements that serve our curriculum and support our hypothesis. Considering the primary purpose of STEPP, our scaffolding environment, and our target users (i.e., students and teachers), the following list of system requirements was compiled:

Table 1: System Requirements

R1. The system must be able to let the user model physics problems in the three STEPP modules as finite state machines (FSMs).
R2. The user can create, remove, and update any state or transition in the FSM of an object.
R3. The system must supply template states, and the user can create and save their own template states.
R4. The system must be able to let the user save and load FSMs in files.
R5. The system must be able to animate the movement of objects described by their FSMs.
R6. The user can play, pause, and reset the animation at any time.

R7. The system can display the graphs of physics variables.

R8. The user can turn on and off the display of the graphs, and resize/relocate the graphs.

R9. The FSM, the animation, and the graphs of an object must be synchronized in “simulation time” which is not real-time and is scaled so that the simulation is completed in a reasonable amount of time.

4 Architecture

The STEPP modules are built in the Unity platform. Unity modules can be deployed either as standalone systems or web-based applications. Since our goal is to provide easy access to STEPP with minimal hardware requirements, the STEPP modules will be deployed through a web application.

STEPP will follow the basic client-server architecture, in which many clients request and receive service from a centralized server. Client computers will be able to access the web application through a browser (See Figure 1). Our clients are both students and teachers. Although they will be accessing the same system, teachers will be provided with additional options. The server will host a database in which information about users and further statistics will be stored and maintained. In addition to providing access the Unity program, the website will post the latest updates about the project. The website will also contain information about the team members and research work and show images that will help visualize the project.

The logs generated from Unity will be stored in a database that will allow us to run further statistical analysis and get a better understanding of application usage. This will help us appreciate how students are utilizing each individual module. The detailed report about usage and test results will help teachers better design the course content and further refine the curriculum.

The app server (See Figure 1) can be either hosted using the Apache HTTP server or any cloud-based server, such as Amazon Web Service or Microsoft Azure.

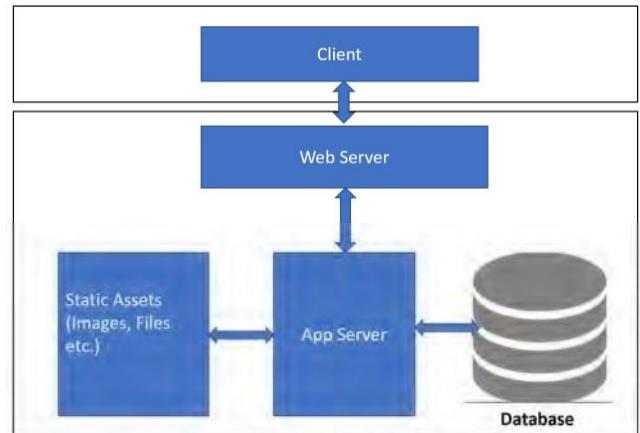


Figure 1: STEPP Architecture

5 STEPP Modules

In this section, we describe some preliminary activities that will prepare students to use the STEPP simulation tools, give a brief overview of learning objectives STEPP is designed to achieve, and present our scaffolded approach to implement the three STEPP learning modules.

5.1 Preliminary Activities

Prior to the introduction of FSMs or STEPP, students will participate in a Module 0 concept mapping activity. Concept maps in the context of a science classroom are tools to provide windows into student thinking [6]. Creating a concept map helps students organize their ideas about the relationships between particular topics. Ideas in a concept map are typically organized relative to a central topic via an interconnected web. Lines connecting nodes of the web often have descriptors of relationships.

Students may or may not have encountered concept mapping prior to the activity. The lesson introduction will therefore provide students with a variety of example concept maps unrelated to concepts in Module 0. The types of information and ideas that a concept map might entail are vast. As a lead in to STEPP, students working in small groups will be tasked with creating concept maps based on the motion of a toy or model car. An example concept map is shown in Figure 2. Prior to diagramming their thinking, students will have an opportunity to interact with the car(s). For the concept map itself, each group will be prompted by the teacher to map their ideas on the motion of the car. Students will be asked to consider multiple scenarios, including no motion, motion at a constant speed, and changes in motion.

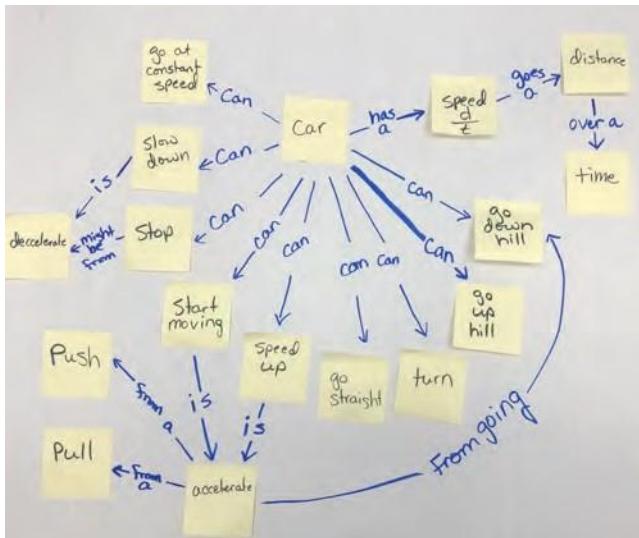


Figure 2: Shown is an example concept map on the motion of a car such as a typical student might create in an introductory physics course in high school. The terminology is natural language as would be expected for a novice. Relationships between nodes represent common beginning student ideas.

The Module 0 activity will be conducted at the beginning of the school year immediately preceding an introduction to kinematics. We therefore anticipate that Module 0 will serve as: (1) an opportunity for an evaluation of some of the initial ideas of students in regard to kinematics prior to formal physics instruction, and (2) the beginning of a process to move students from natural language into formalized physics vocabulary. For the purposes of assessment, teachers will be provided with examples of concept maps reflecting understanding of the physics of the motion of toy cars ranging from a novice to advanced level. The entire lesson Module 0 lesson is designed to fit within one 50minute class period.

Module 0.5 will be short extension of Module 0 to serve as a bridge between the open ended relational form of concept mapping and the specific structure of FSMs. Current conceptualizations of Module 0.5 are in the form of a 5 minute or less video that may be shown in class, or assigned for out-of-class viewing. FSMs, unlike concept maps, involve a progression of states with discrete transitions between them. Module 0.5, currently in development, will provide students with one or more examples of how a subset of descriptors of motion and changes in motion of a toy car, using natural language, can be used to build a FSM (Figure 3). Once the connection between the motion of a car and the formalism of a FSM is made, students will progress to the STEPP learning modules.

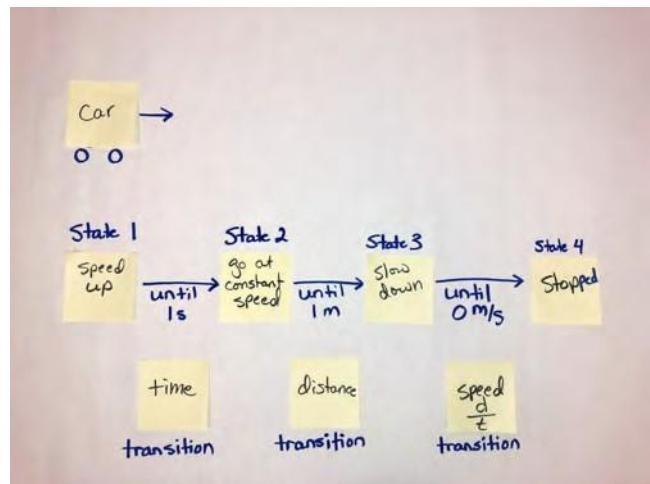


Figure 3: Selected elements of the concept map from Figure 2 have been reorganized into a FSM for 1D motion. The reorganization shown reflects the conceptualization of the FSM-based simulations for STEPP, beginning with simple motion and the natural language of a physics novice.

5.2 Overview of Learning Objectives

The primary goal of STEPP is to increase high school students' understanding of physics concepts and CT through construction of their own simulations. To accomplish this, we develop three STEPP learning modules. The design of these modules is based on prior research and their content is aligned with the Texas Essential Knowledge and Skills (TEKS) [16]. Although the topics of these

modules were selected from TEKS, they are also aligned with the Performance Expectations associated with the topic “Forces and Interactions” of NGSS which “support students’ understanding of ideas related to why some objects will keep moving, why objects fall to the ground ...” [1]. As these topics are taught in other states, the STEPP modules can be readily adapted to physics curricula nationwide. Each STEPP module corresponds to a specific learning objective derived from high school physics TEKS:

Module I: Objective I. One-dimensional kinematics: Describe and analyze motion in one dimension using equations with the concepts of distance, displacement, speed, average velocity, instantaneous velocity, and acceleration (TEKS PHYS.4B).

Module II: Objective II. Two-dimensional kinematics: Analyze and describe accelerated motion in two dimensions using equations, including projectile and circular examples (TEKS PHYS.4C).

Module III: Objective III. Newton’s laws of motion: Calculate the effect of forces on objects, including the law of inertia, the relationship between force and acceleration, and the nature of force pairs between objects (TEKS PHYS.4D).

All three STEPP modules employ scaffolded learning. Using the Unity-based STEPP system, students will build executable statebased models (i.e., FSMs) of physical systems consistent with the physics concepts introduced in each module.

5.3 Implementation

In addition to the requirements (Sec. 3), we also have guidelines and constraints in operating systems, technology, data collection, performance, and security. After forming the specifications, it is essential for us to formalize the physics problem to determine how to model the problems as finite state machines, and how to calculate unknown variables in the problems. Note that a method to model general physics problems as finite state machines is too broad to discuss in this paper. To stay in the scope of this paper, we only show the main principles to formalize physics problems within STEPP’s target: 1D/2D kinematics and Newton’s laws of motion in high school physics. Next the formalization is translated into programming domain from concept art to user interface design, and from software architecture to actual coding implementation. We design our system as a Model-View-Controller (MVC) architectural pattern [17]. The model is the backend of the system which handles all physics calculation and checking. The view is the graphical user interface (GUI) where the system takes input from the user through graphical interaction with a mouse and a keyboard. The controller are the C# scripts which receive the request from the view, manipulate the model, and send the result back to the view. Our implementation aims to meet all the targeted requirements while having high usability, low maintenance, and no bugs. The rest of this section is divided into subsections which will describe each of our three modules in detail.

5.3.1 Module I: One-dimensional kinematics. This completed module is divided into four separate levels based on our scaffolded approach. An example with a chicken character object will be used

to describe the module in the paper (See Figure 4), although students can choose a different character object, such as a car or a train.



Figure 4: Chicken in Module I One-dimensional kinematics

Level 1 provides students with pre-built states that implement motion described in natural language. Students will be able to string these states together into a FSM and then run this FSM to produce a simulation of a one-dimensional system. This level has two learning objectives: (1) to show students the connection between the one-dimensional motion they studied in Module 0 (described in Sec. 3.1) and STEPP simulations, and (2) to illustrate how STEPP realizes the concept of FSMs introduced in Module 0.5.

Level 2 introduces the physical concept of displacement, the vectorial change in the position of an object between the initial time and final time. The total distance traveled by an object is the sum of the magnitudes of the individual displacements. States in Level 2 will be described by their displacements, i.e., students will be able to enter a single real number to assign the displacement of each state. (See Level 2 in Figure 7.) Transitions between states will be implemented automatically after each displacement is completed. The speed will be fixed in the simulation, so the time spent in each state will be proportional to the magnitude of the displacement. Students will be able to watch a graph of the position as a function of time as the simulation occurs.

Level 3 introduces the physical concept of instantaneous velocity, the time rate of change of displacement. Speed is the magnitude of the instantaneous velocity, while the average velocity is the total displacement over a time interval divided by the length of that interval. States in Level 3 are described by their instantaneous velocity, which is constant within each state but can differ from state to state. Students will need to specify the initial time and position at the start of the first state of the FSM, the time or position at which transitions between states occur, and the time or position at the end of the final state. In addition to the physical concepts of velocity and speed, this level also teaches important CT skills by introducing the Boolean logic that governs the state transitions which are no longer automatic. The system undergoes a transition between states (a change in instantaneous velocity) if the condition specified for that event is satisfied. These conditions

may not be satisfied, such as if a state with negative instantaneous velocity is assigned to end when its final position is larger than its initial position. Students will be given helpful hints when transitions fail to be realized, teaching them to “debug” models of nonphysical behavior corresponding to FSMs that fail to reach the final state. Students will be able to watch graphs of the position and instantaneous velocity as functions of time as these simulations occur; the velocity is stepwise constant while the position is a sequence of continuous line segments.

Level 4 introduces the physical concept of acceleration, which is the rate at which velocity changes with respect to time. States in Level 4 are described by their acceleration, which is constant within each state but can differ from state to state. Students will need to specify the initial time, position, and velocity at the start of the first state of the FSM, the time, position, or velocity at which transitions between states occur, and the time, position, or velocity at the end of the final state. In addition, students will have the option to implement an impulse or “kick” that changes the velocity at a transition between states. Such kicks were automatic in Level 3, where the states were specified by different velocities, but in level 4 we assume that the velocity is continuous (linear momentum is conserved at transitions) unless a kick is explicitly specified. The more complicated motion (nonzero acceleration) and greater diversity of conditions at the transitions implies that the logic that determines whether transitions are achieved is more advanced in Level 4 compared to Level 3, helping the students to further improve their CT skills. Students will be able to watch graphs of the position, velocity, and acceleration as functions of time as these simulations occur. The acceleration is stepwise constant, the velocity is a sequence of line segments that are discontinuous at transitions with nonzero kicks, and the position is a sequence of continuous parabolic segments. We formalize 1D kinematics physics problem as finite state machine. The n^{th} state $S^{(n)}$ begins at a time $t_i^{(n)}$ and ends at a time $t_f^{(n)}$. It begins with a position $x_i^{(n)}$ and velocity $v_i^{(n)}$, and ends with position $x_f^{(n)}$ and velocity $v_f^{(n)}$. It has an acceleration $a^{(n)}$. The system is in state $S^{(n)}$ at times $t_i^{(n)} < t < t_f^{(n)}$, during which its position and velocity are given by:

$$x^{(n)}(t) = x_i^{(n)} + v_i^{(n)}(t - t_i^{(n)}) + \frac{1}{2}a(t - t_i^{(n)})^2 \quad (1) \quad v^{(n)}(t) = v_i^{(n)} + a(t - t_i^{(n)}) \quad (2)$$

All states $S^{(n)}$ must be assigned an acceleration $a^{(n)}$. The first state $S^{(1)}$ must be initialized with the three initial values $t_i^{(1)}$, $x_i^{(1)}$, and $v_i^{(1)}$. The system will transition from state $S^{(n)}$ to state $S^{(n+1)}$ at transition $T^{(n)}$. This transition will occur when one of three conditions specified by the user, $t^{(n)} = t_f^{(n)}$, $x^{(n)} = x_f^{(n)}$, or $v^{(n)} = v_f^{(n)}$, is met. A transition could also include an external force which explicitly set the initial velocity of the next state to be a certain value. As we must know one of the three variables t_f , x_f , or v_f , from user input and all the three initial variables x_i , v_i , t_i from inheritance between states or user input, we could derive the remaining unknown two variables of the three t_f , x_f , or v_f based on the formulas (1) and (2). That step is automatically processed by the STEPP system. As the user enters enough information, the system will immediately

calculate the unknown variables. The STEPP system also automatically checks for user input to validate if a state is physically viable. After calculating and checking, if the FSM is valid, the system can play the animation according to the FSM.

Above all, we are going to take a one-dimensional kinematics question in Level 4 of Module I as an example:

Question description:

In a backyard, a chicken walks to the right in a straight line at an initial speed of 10 cm/s. The chicken slows down and after traveling 25 cm comes to a stop. At the moment the chicken comes to rest, a naughty kid kicks the chicken to the left so that it instantly starts walking at a speed of 20 cm/s. The chicken slows down at the same rate as previously and eventually comes to rest.

- Find the acceleration at which the chicken slowed down.
- Find the total distance the chicken traveled.
- Find the total displacement of the chicken.

State-based solution in STEPP:

According to above question description, in STEPP, students need to build a Finite State Machine (FSM) to model this question, afterwards they play the simulation, that is, an animation. Inside the FSM of the chicken, there should be two states S1 and S2: S1 tackles the state of descriptive scenario of “a chicken walks to the right in a straight line at an initial speed of 10 cm/s. The chicken slows down and after traveling 25 cm comes to a stop.” while S2 tackles the state of descriptive scenario of “At the moment the chicken comes to rest, a naughty kid kicks the chicken to the left so that it instantly starts walking at a speed of 20 cm/s. The chicken slows down at the same rate as previously and eventually comes to rest.” Also there should be a conditional transition between these two states to handles events like “traveling 25 cm” and “Kick.” In STEPP, we designed a formalized variable table for all variables, e.g., $t^{(n)}$, $x^{(n)}$, $v^{(n)}$ and $a^{(n)}$, of each state or transition, as shown at the right lower corner in Figure 5.



Figure 5: Chicken example question in Level 4 of Module I.

The variable tables of the states S1 and S2 and the transition between them are shown in Figure 6.

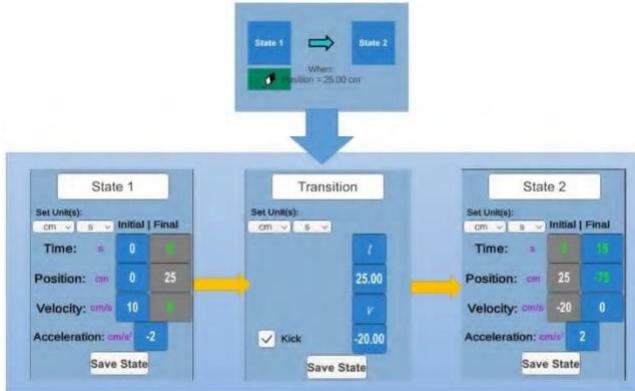


Figure 6: Formalized variable tables of the states and transition in the chicken example question in Level 4 of Module I

In Figure 5, the FSM menu panel is seen in the left lower corner. The middle bottom area is the FSM build panel. In the upper animation scene, where the chicken is standing at the final position of -75 cm, the chicken's acceleration is visualized by the orange animated arrow. (The chicken's velocity would be visualized by the green animated arrow if the chicken had a nonzero velocity). The three graphs that plot chicken's displacement, velocity, and acceleration over time in "simulation time" are displayed in the area as well. With synchronized animation and graphs, students can see the turning point at which chicken's motion changes and the difference between the total distance and total displacement of the chicken, where the physical concepts of distance, displacement, speed, velocity, and acceleration are required by TEKS 4B.

In STEPP Module I, Levels 2, 3 and 4 have similar GUIs, except for the differences in their formalized variables tables for states and transitions. The variable tables are scaffolded, based on the physical concepts introduced in each level. In Level 1, there is no need for students to fill out formalized variable tables since we provide pre-built states that implement motions described in natural language. For instance, in Figure 3, students are asked to string the states of a car object together into a FSM and then play it to produce a one-dimensional simulation of the motion of the car to the right. Students could alternatively have chosen the motion to be directed to the left. In this level, the conditions in transitions are set by time by default. Details are in Figure 7.

5.3.2 Module II: Two-dimensional kinematics.

STEPP Module II is organized into four scaffolded levels similar to Module 1.

Level 1 introduces displacement as a two-dimensional vector, but is otherwise identical to Level 2 of Module I. Each state of the FSM is characterized by a vectorial displacement that can be specified by either its Cartesian x and y components or its magnitude and direction (specified by the angle it makes with respect to the x axis).

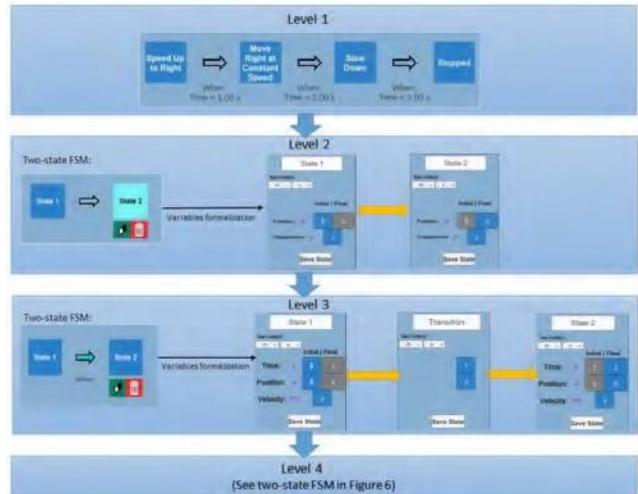


Figure 7: Established scaffolding of levels in STEPP Module I

The final position of a given state is the vector sum of the initial position and the displacement of that state. Transitions are automatic, with the initial position of a state equal to the final position of the proceeding state. The simulations will depict motion in the xy plane as viewed from above. The speed will be constant throughout the simulation, so the duration of each state will be proportional to the magnitude of its displacement. Students will be able to view the x and y components of the position as functions of time. This level has three learning objectives: (1) to introduce the top-down view of simulations in two dimensions, (2) to teach students about vector addition, and (3) to review the basic trigonometry needed to translate between the expressions for a vector in Cartesian components or by magnitude and direction, i.e. the Pythagorean theorem and the SOHCAHTOA trigonometric identities.

Level 2 introduces the two-dimensional instantaneous velocity. States of the FSM will be characterized by constant velocities (as in Level 3 of Module I) which we allow to be specified by either their Cartesian x and y components or their magnitude and direction. The speed is the magnitude of the velocity. The final position of a state is the vector sum of the initial position and the product of the velocity and time interval (the difference of the final and initial time). The transitions are no longer automatic but occur when either the time or the x or y component of the position attain a final value that must be explicitly specified. Kicks are automatically implemented at transitions because the velocity changes between states. Students can view graphs of the Cartesian components of the position and velocity as the simulation occurs.

Level 3 introduces two-dimensional accelerations. States of the FSM will be characterized by constant accelerations (as in Level 4 of Module I) which we allow to be specified by either their Cartesian x and y components or their magnitude and direction. States with accelerations of -9.8 m/s^2 in the y direction can be used to simulate projectile motion. Transitions can now be specified by either a time or Cartesian component of the position or velocity.

Kicks at transitions are now optional and must be specified explicitly if an impulse (non-conservation of linear momentum) is desired at these transitions. Students can view graphs of the Cartesian components of the position, velocity, and acceleration as the simulation occurs.

Level 4 introduces states in which objects undergo uniform circular motion. The acceleration in these states is constant in magnitude but not direction, as it always points towards the center of the circle. Once students specify the magnitude of the centripetal acceleration and whether the circular motion is clockwise or counterclockwise, the initial position and velocity determine the center of the circle, its radius, and the speed at which the object moves. Transitions out of a state of circular motion occur at either a specified final time or angular position. As in Level 3, optional kicks are allowed at transitions and students can view graphs of the Cartesian components of the position, velocity, and acceleration as the simulation occurs.

The 4 levels of Module II address in a scaffolded sequence motion in two dimensions including projectile and circular motion as required by TEKS 4C. Figure 8 presents the draft graphic user interface (GUI) of Module II.

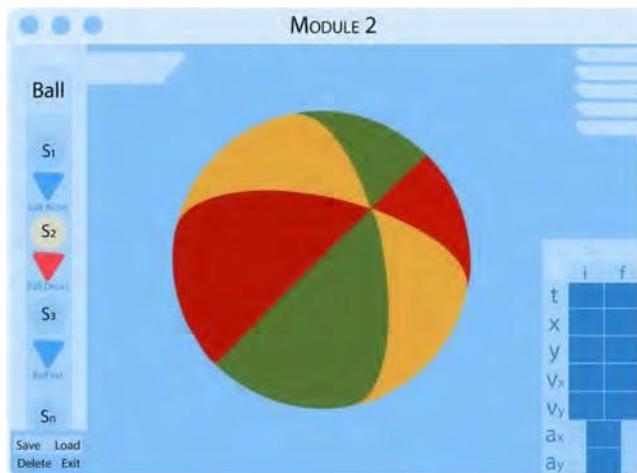


Figure 8: GUI design of STEPP Module II

5.3.3 Module III: Newton's Laws of Motion. STEPP Module III focuses on targeted opportunities to develop student conceptual understanding of Newton's Laws of Motion. The module curriculum and suite of simulations will build upon both the motion concepts introduced in the prior models and the concepts of FSMs. The idea of transitions between states in a FSM will be leveraged to guide students in moving from common incorrect ideas in physics into Newtonian thinking. Module III will also target specific areas of known difficulty for students in the development of Newtonian mental models in the context of uniform circular motion and collisions. Conceptual difficulties in these areas are well documented in a number of studies, and reflected in the *Force Concept Inventory* [18].

A common initial idea for students is that motion requires a constant force or impetus [19,20,21,22]. This is counter to Newton's first law which states that an object will remain in uniform motion (constant velocity) if there are no forces acting on it. Newton's second law states that if there is a net force F acting on an object, its velocity will not remain constant but instead experience a time rate of change (acceleration a) proportional to the net force and inversely proportional to the mass m : $F = ma$.

The first two levels of STEPP Module III will consist of states experiencing no net force and thus uniform motion consistent with Newton's first law. The first level will focus on one dimensional motion (like Level 3 of Module I), while the second level will generalize to two dimensions (like Level 2 of Module II). However, unlike these levels of Modules I and II in which kicks were implemented automatically, in Module III the student must specify the net force acting over a short but finite time interval Δt during the transition. This corresponds to an impulse such as would occur when a ball bounces off the floor or wall. This net force will lead to an acceleration and thus change in velocity that is proportional to the net force and inversely proportional to the object's mass consistent with Newton's third law. STEPP Module III will thus help debunk the common misconception that forces are required to maintain the uniform motion of the states in Levels 1 and 2; a net force only acts during the transitions where changes to the velocity occur.

Levels 3 and 4 of Module III will consist of states with nonzero acceleration in one and two dimensions respectively, like Level 4 of Module I and Level 3 of Module II. However, rather than directly specify the acceleration as in Modules I and II, the students will now specify the mass of the object and each of the forces acting during each state. STEPP will add these forces to determine the net force (using vector addition in Level 4), then determine the acceleration from the mass and net force using Newton's second law. Students using Levels 3 and 4 of Module III will be guided to develop an intuitive understanding of Newton's second law including such important realizations as that the velocity (and thus linear momentum) is conserved in directions perpendicular to the net force (such as perpendicular to an inclined plane or in the horizontal direction in the case of projectile motion). Arrows representing each force vector and the net force (parallel to the acceleration) will be added to the animations in Module III.

Level 5 of Module III will include states of uniform circular motion as in Level 4 of Module II. Circular motion is constrained motion and is challenging for students [18,21]. Although the centripetal acceleration is inward in these states, if the inward force of constraint is removed (such as the string of a sling suddenly snapping), the object will continue its motion in a straight line tangent to the circle consistent with Newton's first law. Students will often erroneously assume that the object will move radially outward due to a fictitious "centrifugal" force. Level 5 of Module III will help students realize that this outward motion is nonphysical; the removal of the constraint is a transition between

states with radial and zero acceleration, but the tangential velocity is continuous through the transition since no explicit impulse is applied.

Level 6 of Module III will focus on collisions in one dimension. This is an ideal system for studying Newton's third law of motion, that if an object exerts a force on a second object, that second object will exert a force on the first object that is equal in magnitude but opposite in direction. Students will specify masses and initial velocities for two objects; in one dimension, as long as the relative velocity is negative, the two objects will inevitably collide (beads constrained to move on a necklace cannot pass through each other). We will focus on elastic collisions (in which energy is conserved), but future versions of STEPP may also allow inelastic collisions including fully inelastic collisions in which the two objects stick together. Students will be able to freeze the simulation at the instant of the collision to see that the forces the two objects exert on each other are equal and opposite as indicated by arrows of equal length and opposite direction designating the force pair. This implies that the total linear momentum is conserved during the collision since the sum of the forces acting on the two objects is zero. However, each object experiences an acceleration and thus change in velocity inversely proportional to its mass consistent with Newton's second law (light objects experience bigger velocity changes like a ping pong ball hitting a paddle). Students will learn this from both the acceleration arrows (which will be opposite in direction but unequal in magnitude) and the subsequent motion post-collision. Through a variety of simulation scenarios in Level 6, students will come to understand the difficult concept that the equality of the magnitude of the forces required by Newton's third law is entirely independent of the motion and masses of interacting objects. The draft conceptual graphic user interface (GUI) of Module III presented in Figure 9.

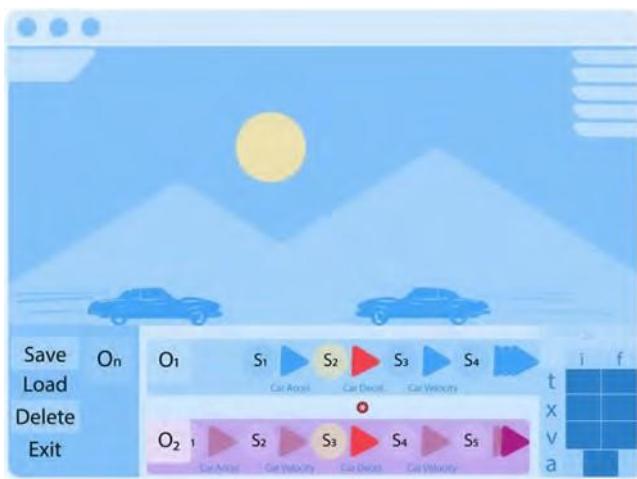


Figure 9: Conceptual GUI design of STEPP Module III

6 Assessment Plan

This project tests the hypothesis that instruction in physics and computational thinking (CT) can be synergistic, i.e. that students

can learn these subjects better when they are taught together rather than separately. We will assess the effectiveness of the STEPP modules at high schools in the Dallas-Ft. Worth metropolitan area. Physics classes will be randomly assigned to one of three experimental groups: a control group with the standard physics curriculum, a group who uses only pre-built STEPP simulations, and a group who will build their own simulations within the framework of the STEPP modules. In an effort to minimize variability in the data, the team has recruited teachers with multiple sections of on-level physics and each teacher will implement more than one of our experimental conditions. Formative assessments in physics learning will be developed in collaboration with high school teachers. Summative assessment of students' gain in physics conceptual understanding will be measured by the *Force Concept Inventory* [18], an instrument in widespread use in the physics education research community. Gains in CT, state-based modeling, and programming concepts will be assessed with measures developed by the research team and in collaboration with high school teachers. Changes in students' attitudes towards computing will be measured by the Computing Attitudes Survey [23]. We will also collect usability feedback from both teachers and students and measure student demographic information to explore whether there are individual differences (e.g., sex, ethnicity) in the effectiveness of the STEPP modules in facilitating gains in physics knowledge, CT, and attitudes towards computers. Finally, unique to the experimental conditions utilizing STEPP, our research team will track students' module use and their actions so that we can conduct a fine-grained analysis of how students' different actions while using the modules predict gains in physics knowledge and CT.

7 Conclusion

The Scaffolded Training Environment for Physics Programming (STEPP) is currently being developed by an interdisciplinary team of faculty and students from programs in Arts, Technology, and Emerging Communications, Computer Science, Physics, Psychology, and Science/Math Education. This project, funded by the NSF STEM + Computing Program, is premised on the hypothesis that instruction in physics and computational thinking (CT) can be synergistic. The three STEPP modules currently in development (one-dimensional kinematics, two-dimensional kinematics, Newton's laws of motion) are aligned with three specific learning objectives of the Texas Essential Knowledge and Skills (TEKS) [16]. This same physics curriculum is also a part of the Next Generation Science Standards (NGSS) which has been adopted throughout much of the United States.

STEPP teaches CT primarily through the use of finite-state machines (FSMs) for modeling and simulation (M&S) of the motion of physical systems. FSMs are widely used for M&S in engineering and computer science but are not included in traditional physics curricula. Perhaps this is because classical physical systems evolve through continuous rather than discrete physical states. However, the continuous trajectories of systems can be broken into discrete segments that can be regarded as the sequential states of a FSM.

This approach is particularly useful for teaching introductory physics to high school students because: (1) in algebra-based physics, the acceleration is piecewise constant, implying that the states of our FSMs have physical significance beyond arbitrarily defined time intervals, and (2) the episodic sequence of states in a FSM more closely aligns with students' precalculus intuition about how physical systems behave.

The use of FSMs, coupled with the scaffolded approach of STEPP, allows us to provide students with simulations that are physically accurate yet initially described by natural-language states in accord with their intuition. As physical concepts like displacement, velocity, and acceleration are introduced at increasingly higher levels of each module, students will learn how the intuitive states of motion can be more rigorously defined in the language of physics. This rigor can then be used to demonstrate that common physical misconceptions (like

Aristotelian notions of coming to rest, fictitious centrifugal forces, etc.) are ill posed for systems constructed from the physically allowed states of our FSMs.

This hypothesis that STEPP can successfully teach physics and CT synergistically will be critically tested later this year. After completing beta versions of the three STEPP modules in Spring 2019, we will recruit local high school teachers to attend a summer institute. The participants will be trained in the use of STEPP and work with our research team to develop curricula appropriate for their own classrooms. These teachers, monitored by our assessment team, will then deploy STEPP in on-level physics classrooms in Fall 2019. The performance of students building their own simulations using STEPP will be assessed using the Force Concept Inventory [11] and Computing Attitudes Survey [23], then compared with control groups either passively watching similar simulations or using the traditional local physics curriculum. If the active STEPP approach proves successful in teaching physics and CT synergistically, we will apply to the NSF and other sources for funding to develop additional STEPP modules and expand the curriculum to other STEM disciplines. We hope that STEPP can live up to its name as an important step towards incorporating the M&S mindset into students at the K-12 level, and that this early exposure to M&S will continue to bear fruit during their university education and eventual STEM careers.

ACKNOWLEDGMENTS

The project is supported by the National Science Foundation under Grant No. 1741756. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

- [1] NGSS Lead States. *Next Generation Science Standards: For States, By States*. Retrieved from <http://www.nextgenscience.org>.
- [2] J. M. Wing (2006). *Research Notebook: Computational Thinking—What and Why?* Retrieved from <https://www.cs.cmu.edu/link/research-notebookcomputational-thinking-what-and-why>.
- [3] B. J. Reiser (2004). Scaffolding Complex Learning: The Mechanisms of Structuring and Problematising Student Work, *The Journal of the Learning Sciences*, 13, 273-304.
- [4] J. D. Novak (1990). Concept Mapping: A Useful Tool For Science Education. *Journal of Research in Teaching*, 27(10), 937–949.
- [5] L. Austin and B. Shore (1995). Using concept mapping for assessments in physics. *Physics Education*, 30(1), 41–45.
- [6] J. Vanides, Y. Yin, M. Tomita, and M. Ruiz-Primo (2005). Using Concept Maps in the Science Classroom. *Science Scope*, 28(8), 27–31.
- [7] J. Nesbit and O. Adesope (2006). Learning with Concept and Knowledge Maps: A Meta-Analysis. *Review of Educational Research*, 76(3), 413–448.
- [8] T. S. Chow (1978). Testing Software Design Modeled by Finite-State Machines. *IEEE Transactions on Software Engineering*, SE-4(3), 178-187.
- [9] C. E. Shannon (1948). A Mathematical Theory of Communication. *The Bell System Technical Journal*, 27, 79–423, 623–656.
- [10] D. Harel (1987). Statecharts: a visual formalism for complex systems. *Science of Computer Programming*, 8, 231–274.
- [11] W. Christian and M. Belloni (2003). *Physet Physics: Interactive Illustrations, Explorations and Problems for Introductory Physics*. Addison-Wesley Publishing. Retrieved from <https://www.compadre.org/physets/>.
- [12] A. Paul, N. Podolefsky and K. Perkins (2012). Guided without feeling guided: implicit scaffolding through interactive simulation design. *Proceedings of the 2012 Physics Education Research Conference*, 1513, 302-305.
- [13] P. Fishwick (1994). *Simulation Model Design and Execution: Building Digital Worlds*, Prentice Hall.
- [14] P. Fishwick, Ed. (2007). *Handbook of Dynamic System Modeling*, Chapman & Hall. CRC Computer and Information Science Series.
- [15] S. Robinson (2015). A tutorial on conceptual modeling for simulation. *Proceedings of the 2015 Winter Simulation Conference*, Eds. L. Yilmaz., W. K. V. Chan, I. Moon, T. M. K Roeder, C. Macal, and M. D. Rosetti. IEEE. Piscataway, NJ. 1820–1834.
- [16] Texas Education Agency. *Texas Essential Knowledge and Skills*. Retrieved from <http://tea.texas.gov/index2.aspx?id=6148>.
- [17] G.E. Krasner and S.T. Pope (1988). A cookbook for using the Model-View-Controller user interface paradigm Smalltalk-80. *Journal of Object-Oriented Programming*, 1(3), 26-49.
- [18] D. Hestenes, M. Wells, and G. Swackhamer (1992). Force Concept Inventory. *The Physics Teacher*, 10(3), 141–158.
- [19] J. Clement (1982). Students' preconceptions in introductory mechanics. *American Journal of Physics*, 50(1), 66–71.
- [20] I. Halloun and D. Hestenes (1985). Common sense concepts about motion. *American Journal of Physics*, 53(11), 1056-1065.
- [21] R. Thornton and D. Sokoloff (1998). Assessing student learning of Newton's laws: The Force and Motion Conceptual Evaluation and the Evaluation of Active Learning Laboratory and Lecture Curricula. *American Journal of Physics*, 66(4), 338-352.
- [22] S. McLaughlin (2006). Rounding Up Students' Conceptions on Circular Motion. *Iowa Science Teachers Journal*, 33(2), 7-15.
- [23] B. Dorn and A. E. Tew. *Computing Attitudes Survey*. Retrieved from <http://faculty.ist.unomaha.edu/bdorn/cas.html>.