11-9-2017

Processing Motion: Using Code to Teach Newtonian Physics

M. Ryan Massey
mrm453@mail.umsl.edu

Processing Motion: Using Code to Teach Newtonian Physics

M. Ryan Massey

A Dissertation Submitted to the Graduate School at the University of Missouri-St. Louis
In partial fulfillment of the requirements for the degree
Doctor of Philosophy in Education

December
2017

Advisory Committee

Keith Miller, Ph.D.
Chairperson

Cody Ding, Ph.D.

Charles Granger, Ph.D.

Gualtiero Piccinini, Ph.D.

Abstract

Prior to instruction, students often possess a common-sense view of motion, which is inconsistent with Newtonian physics. Effective physics lessons therefore involve conceptual change. To provide a theoretical explanation for concepts and how they change, the triangulation model brings together key attributes of prototypes, exemplars, theories, Bayesian learning, ontological categories, and the causal model theory. The triangulation model provides a theoretical rationale for why coding is a viable method for physics instruction. As an experiment, thirty-two adolescent students participated in summer coding academies to learn how to design Newtonian simulations. Conceptual and attitudinal data was collected using the Force Concept Inventory and the Colorado Learning Attitudes about Science Survey. Results suggest that coding is an effective means for teaching Newtonian physics.

Acknowledgements

My graduate studies were funded in part by CISTL Cadre, the UMSL Advanced Credit Program, the Maplewood-Richmond Heights School District, and my mom, Sandra Kane.

Thank you, James Tipton, for taking AP English and knowing how to make my references APA compliant. Thank you, Mia Rintoul and Henry Seymour, for being amazing students and serving as tech aides at the coding academies.

I would like to acknowledge a short list of my teachers:

- Richard Picard, high school physics teacher at Parkview Science Magnet
- A guy who sat next to me in AP Calc, who taught me how to program the TI-82
- Brant Hinrichs, extraordinary physics educator at Drury University
- Mark Wood, for teaching me Fortran and more physics than chemistry at Drury
- Joseph Polman, for teaching me about little scientists and project based learning
- Robert Northcott, for inspiring me to study philosophy
- Cody Ding, professor of quantitative analysis and possessor of great style
- Gualtiero Piccinini, who taught me about the mind and how to write a thesis
- Bill Kyle, who taught me how to write a dissertation
- Charles Granger, for showing me and countless others how to be a science teacher
- Keith Miller, for your essential support and exceptional advice

The Processing Motion project would not have been possible without the students and staff from the participating schools. It was a great privilege to work with you.

Thank you, Sylvia Massey, for frequently shutting my laptop and asking me to tell you stories about the mysterious planet Merf. I tried to oblige you often enough so that you would not notice I was writing a dissertation. And thank you Mimi, for playing bunnies, when Merf stories had to wait.

Table of Contents

Chapter 1: Introduction & Theory

Much progress has been made towards understanding how concepts function in particular contexts. The Processing Motion project applies concepts research to a new methodology for physics education. I will not describe every theory here, since this is not a big book of concepts, but such a book does exist (Murphy, 2002). Although the empirical research on concepts tends to be fairly technical, an underlying theme of the current study is to synthesize the results into a format that can be readily utilized by classroom educators. After all, classroom teachers do not necessarily need to worry about how to evaluate the Minkowski r-metric for the i^{th} and j^{th} stimuli within the k^{th} category dimension. But see Nosofsky (1986) if you are one of the teachers who does worry about such things; he provides a great explanation. The primary goal of my project is to provide a strong theoretical approach to learning and to contribute data to the existing body of evidence pertaining to physics education.

Significance of Study

In 1980, Seymour Papert proposed giving students in the class of 2000, who would be in kindergarten in 1987, their own personal home computer (Papert, 1980). These students would be taught to code and their programs would provide a new medium for representing what they learned in the content areas. To paraphrase Papert, we want the students to program the computer, rather than the computer to program the students. As a member of the class of 2000, I can confirm that Papert's proposal was not broadly implemented. Today, computers are in schools and computer applications are used frequently by students and teachers. However, Papert's educational coding dream has generally not come true. As we look for ways to improve physics instruction, it might be fruitful to revisit Papert's ideas.

Problem Statement

Despite considerable effort from educators and researchers, conceptual change continues to be a major obstacle, both in classrooms and experimental settings. Various phenomena have been identified pertaining to conceptual change. Various theories have been developed to posit causes and predict effects. This study aims to consolidate a subset of the empirically successful theories related to concepts and conceptual change. The resulting approach, which I call the triangulation model, helps explain the effects emphasized by various concept theories and served as a framework to guide my educational methodology. Broadly speaking, conceptual change is initiated by the triangulation of prior knowledge, hypothesis, and observation as described by Bayesian learning. This is mostly unrelated to the counterfactual snake "triangulation method" found in Fodor (2008, p. 214). As for how to design and implement this teaching methodology, I used the triangulation model to constrain the options.

The resulting lessons emphasize teaching students how to develop their own physics simulations using the Processing coding language.

Theoretical Framework

What follows looks like a literature review, and it is, but the present review is meant specifically to justify my theoretical framework. A traditional lit review in its customary location can be found in Chapter 2.

This study will engage with several theories related to conceptual change. Influenced by Piaget and Kuhn, Posner and his colleagues sought to describe the conditions needed for people to accept a new concept. They suggested that people must be dissatisfied with their current concept and the new concept must be intelligible, plausible, and useful (Posner, Strike, Hewson & Gertzog, 1982). According to Posner et al. (1982), education should focus on experiences that challenge student concepts and then provide or facilitate the discovery of new concepts that make sense and demonstrate utility.

Often citing Posner et al. (1982), a vast literature has developed on the nature of concepts. If we are to understand conceptual change, it would be helpful to have a plausible account of concepts themselves. The next section will look at the pieces vs. coherence debate, which concerns the psychological structure of concepts.

Pieces vs. coherence. On one side of the debate, diSessa has focused on breaking down concepts into smaller constituents. He suggests that concepts are built up from phenomenological primitives, or p-prims, which are gleaned directly from experience (diSessa, 1993). When combined, p-prims allow us to make mechanistic inferences regarding our observations. This "sense of mechanism" includes the ability to assess likelihoods, make predictions, and provide causal descriptions (diSessa, 1993, p. 106). However, p-prims do not naturally form a coherent system (diSessa, 2013). In other words, we might conceptually organize p-prims in ways that are inconsistent or logically incompatible. Overall, diSessa's view is that concepts tend to be independent and fragmented. But there are alternative models with bigger pieces.

Chi and Slotta (1993) propose that concepts are organized into ontological categories. These ontological categories are considered more coherent than diSessa's p-prims. Slotta, Chi, and Joram (1995) suggest conceptual change involves learning a new ontological category or updating the categorization of a concept. And prior ontological commitments could interfere with new conceptual learning. For example, naive physicists are known to place nonmaterial concepts such as force, heat, light, and voltage into substance-based categories (Reiner, Slotta, Chi, & Resnick, 2000). If a student believes heat has the same properties as a liquid, this can result in confusion inside the physics classroom. From this view, teachers should design lessons to prevent and repair categorical mistakes (Chi, 2013).

Vasniadou, Skopeliti, and Ikospentaki (2004) have methodological concerns about how conceptual theories are tested. In a study on astronomical knowledge involving 72 early elementary students, they found the method of questioning significantly impacts the results. In particular, when students are given options to choose, they are relatively good at choosing the scientific answer but their answers are less coherent overall. When the choices are open-ended, student answers are less scientific, but are more consistent. With this in mind, Framework Theory asserts a larger conceptual structure and seeks to explain the cases where incoherence is observed (Vosniadou & Skopeliti, 2013). The central claim is that concepts are organized into a Framework Theory which contains ontological categories plus causal relationships (Vosniadou, 2013). By emphasizing causality in the Framework Theory, Vosniadou has provided a mechanism for how concepts are used to make predictions and explanations. With p-prims, these abilities, including causality, emerge from pieces of concepts. But in framework theory, causation is part of the concept. Causal reasoning therefore plays a more fundamental role in Framework Theory. Framework Theory explains observed incoherence with synthetic concepts (Vosniadou & Skopeliti, 2013). Synthetic concepts occur when logically incompatible concepts, like those from a valid scientific theory, are blended with a naive framework theory. Synthetic conceptions represent a transitional phase between non-scientific and scientific theories. In summary, people develop a framework theory which is internally consistent. When exposed to formal scientific concepts, an incoherent synthetic model is produced, resulting in further misconceptions. If this cognitive dissonance is addressed, by instruction or otherwise, the framework theory can be discarded and replaced by a scientific theory.

Theory theory takes it a step further by claiming that concepts essentially are scientific theories to begin with (Gopnik & Meltzoff, 1997). Before discussing theory theory in detail, I will take a detour regarding another conceptual debate. But first, a nuanced preview.

It is not always clear whether the amount of psychological coherence is a result of personal subjectivity or more objective fuzziness. In a study with 64 undergraduate students, McCloskey and Glucksberg (1978) found that people are less consistent in their usage of a term when the item was less typical of its category. For example, people are consistent with football as a member of the sports category, but are less consistent with chess. I think when researchers focus on smaller pieces of knowledge, typicality effects could be driving the amount of coherence. Less typical items are used more inconsistently, which makes concepts appear less coherent. When viewed from a wider angle, inconsistent treatment of chess might look more coherent, because in some contexts it is sports-like and in other contexts, it is more typical of board games or other non-sport activities. A related issue is the intransitivity of natural categories. Participants in the Hampton (1982) study agreed that car seats are chairs and that chairs are furniture, but did not agree that car seats are furniture. Thus, people were willing to categorize items based on typicality, even when it resulted in counterexamples to their category judgements. These results suggest that many instances of incoherence are not random, but are influenced by the inclination to choose different categorization techniques based on the circumstances.

Likeness vs. likelihood. After reading many literature reviews on concepts, I have identified the common pattern which is to cite Rosch (1975) as the beginning, then gloss over the middle, and conclude with a new theory that is better than the old ones. Since anything before 1975 is rarely mentioned and part of my target audience is classroom teachers who may be unfamiliar with this literature, I am going to go back a bit further.

During the 1950's at Lackland Air Force Base in Texas, Fred Attneave was conducting foundational research on concepts. He was trying to discover if prior "prototype" learning would be transferred to categorization tasks involving new prompts (Attneave, 1957). Attneave conducted two experiments, one with categories of letter pattern variations and one with categories of nonstandard shapes. The results identified familiarity as a key factor in categorization and participants were better at categorizing new variations when the prototype was varied along the same dimensions as their training.

Posner and Keele (1968) expanded Attneave's findings on variation by discovering evidence that prototype concepts are abstracted from learned patterns. Even when participants were not directly shown the prototype used to generate the variations, they still demonstrated categorical reasoning based on the central tendency of the set.

Participants could categorize new patterns just as quickly as patterns they had learned during training, suggesting that familiarity with the abstracted prototype is more important than familiarity with directly learned instances.

The theory that concepts are abstracted prototypes became prominent in psychological literature. Using categories of schematic faces, Reed (1972) tested various quantitative models including Cue Validity, Proximity Algorithm, Average Distance, Prototype, and Weighted Features. See Reed's (1972) appendix for the equations. I have provided a verbal description of each model's categorization rule in Table 1.1. Reed confirmed that abstracted prototypes are used for categorization tasks. Specifically, the weighted prototype model correlated the best with experimental results. However, the weighted average distance method also performed well. In subsequent studies, this model inspired a leading contender, known as Exemplar Theory. Overall, these results provide evidence that people perform categorization tasks by comparing the similarity of an object to category prototypes, but with emphasis on the features that readily distinguish the categories. In other words, category features have unequal influence on determining category membership.

Table 1.1. Models from Reed (1972) Experiment

Model	Category Comparison Rule
Cue Validity	Frequency of category cues
Proximity Algorithm	Distance to nearest category member
Average Distance	Average distance to category members
Weighted Average Distance	Average distance to exemplars with weighted features
Prototype	Distance to average category member
Weighted Features Prototype	Distance to prototype with weighted features

In every model tested by Reed (1972), the category features are quantitatively independent. The equations multiply the weighting factors, but not the variables. Each of these equations are fundamentally based on summation and sums cannot equal zero, unless all the variables are independently equal to zero. The weighting factors help account for typicality effects, where some features are perceived as more typical of category members. However, the weighting factors do not allow for interactions between the feature variables. Medin and Schaffer (1978) contend that independent features pose a serious qualitative risk to the prototype theory. Because in some categorization tasks, no matter how similar the other features, we want a single feature to be able to disconfirm a match. With an additive model, high degrees of similarity in other features can override low similarity relative to a critical feature. To solve this problem, they introduced the “context theory,” which is based on exemplars (Medin & Schaffer, 1978). Context theory assumes that people store specific examples, or exemplars, of category members. New items are categorized based on their similarity to stored exemplars within a category. Medin and Schaffer (1978) used a multiplicative rule to mathematically combine features. As the similarity to an important feature approaches zero, the multiplied term also goes to zero, regardless of similarity within other dimensions. In four experiments involving geometric forms and schematic faces, they found significant, but small quantitative improvements over the prototype theory’s additive model (Medin & Schaffer, 1978). Although the original context model used binary values, it was later generalized by Nosofsky (1986) to allow for continuous variables.

To test models based on similarity, Smith and Minda (1998) performed experiments which asked participants to categorize pronounceable nonsense words and bug drawings. In trials with less category variation, the exemplar models were consistently better. However, in trials with more category variation, the prototype model was favored in early learning and then switched to exemplar model as participants became more familiar with the stimuli. Therefore, people might choose between prototypes and exemplars, depending on the context and their level of familiarity. However, the same researchers conducted a meta-analysis of 30 exemplar studies and found the evidence to be lacking in support of exemplar theory (Smith & Minda, 2000). In response, Nosofsky (2000) discovered irregularities with their meta-analysis, which raised further questions about the empirical state of exemplar theory. Then, using a dot-pattern categorization experiment, Smith (2002) found clear evidence in favor of prototype typicality gradients over exemplar theory. Overall, the exemplar vs. prototype debate has not been decisively settled. There is good evidence for and against both theories.

After gaining some traction with exemplar theory, Douglas Medin partnered with Gregory Murphy in 1985 to produce a revolutionary alternative to similarity based concepts. They suggest that concepts are substantially more coherent than similarity models can accommodate (Murphy & Medin 1985). While similarity theories have accounted for important effects (e.g. typicality), they cannot explain how people choose which features to engage when making category judgments. Also, attribute matching cannot explain the full organizational structure of concepts. Murphy and Medin (1985) assert while similarity fails to explain conceptual coherence, theoretical knowledge is up to the task. By using theories, people can organize concepts based on abstract relationships such as causality. These relationships provide the ability to produce greater coherence within conceptual structures. While Murphy and Medin (1985) specifically distinguish conceptual theories from scientific theories, others suggest that conceptual coherence can rival that of science.

Gopnik and Meltzoff published *Words, Thoughts, and Theories* in 1997. It explained the nature of theories; they are based on abstract notions of coherence, causality, and counterfactuals. Also, theories can be used to predict, interpret, and explain. The central claim is that conceptual development in children is substantially similar to the development of scientific theories. Specifically, concepts and scientific theories share the same structure and function. Theory theory therefore posits an equivalence between conceptual theories and scientific theories.

Initially, theory theory did not have a formal quantitative model, but this changed with the assertion that theoretical learning could be described with a Bayes nets causal model. Theory theory advocates have proposed causal maps, which are mental representations of the causal relationships between objects in the world (Gopnik, Glymour, Sobel, Schulz, & Kushnir, 2004). However, Bayes nets were not powerful enough to account for the ability to rapidly identify causal information from a limited number of observations. Bayesian probabilistic inference, which is formulated from Bayes' Theorem, does appear to capture this ability (Gopnik & Tenenbaum, 2007). The Bayesian form of theory theory also allows for a unification with constructivism (Gopnik & Wellman, 2012). In the past, theory theory had trouble explaining how learning builds on prior knowledge. With Bayesian learning, prior knowledge is built directly into the model.

Bayes' Theorem can be written as $P(H|E) = P(H|K)\,P(E|H) / P(E|K)$, where H = Hypothesis, K = Prior Knowledge, and E = Observed Evidence. It computes the likelihood that a hypothesis is true, given some evidence. Since this learning model typically assumes the probability of the evidence is constant, the denominator is often omitted, as it is simply a normalizing factor. Symbolically, it is $P(H|E) \alpha P(H|K)\,P(E|H)$. Thus, [the probability of the hypothesis given some evidence] is proportional to [the probability of the hypothesis given your prior knowledge] times [the probability of observing the evidence given the hypothesis]. For a better explanation of causal learning, see Pearl (1988) and Pearl (2000).

Theory theory sounds promising, but what is the evidence that concepts include more than simple associations? To start, Waldmann and Holyoak (1992) demonstrated that associative learning is not equivalent to causal learning in a study of university students in Germany. Correlational evidence suggests pre-school children use counterfactual reasoning during pretend play (Buchsbaum, Bridgers, Skolnick, Weisberg, & Gopnik, 2012). And in a study with 64 participants, Schultz and Bonawitz (2007) found that preschool children notice when evidence is confounded and are more likely to explore causally confounded toys than unconfounded toys. Causation appears to play a significant role in adult reasoning too. Rehder and Hastie (2001) found that causal knowledge affects how people perform categorization tasks. Then, Rehder (2003a) performed three experiments to test the way people use causal knowledge with novel categories. The results show support for causal model theory, which claims "people view causal relationships as being constituted by probabilistic causal mechanisms, rather than by relationships of necessity and sufficiency" (Rehder, 2003a, p. 1149). In a related experiment involving 108 university students, Rehder (2003b) compared the configural features prototype and exemplar fragments models with the causal model theory. These models were designed to explain if and if so, how people use causal information to make category judgments in cases where there is a common-cause or common-effect. Causal model theory claims that people judge the category membership of an item based on the likelihood that its features match the pertinent causal mechanism. The Rehder (2003b) results show that causal-model theory correctly predicted there would be no higher-order interactions in the case of common causes and correctly predicted a discounting effect in cases involving common-effects. Thus, when the first cause is paired with an effect, this changes the likelihood equation more than when a second cause is identified for the same effect. In five additional experiments, Rehder (2006) found strong evidence for a causal preference over similarity during categorization tasks. In other words, when people possess causal and similarity based information, they use the causal knowledge to a significantly greater extent. In human reasoning, causation takes precedence over similarity.

The significant accumulation of evidence on all sides of this debate has inspired Machery (2009) to suggest we might want to give up hope for discovering a single psychological mechanism to explain the phenomenon of concepts and that prototypes, exemplars, and theories represent fundamentally different kinds of concepts. In my view, if similarity is part of the mechanism for determining likelihood, a single theory of concepts might still be tenable. Piccinini and Scott (2006) have argued that while it is not wise to assume concepts are a single natural kind, the previous attempts to split concepts have not been fully justified. In a follow up paper, Piccinini (2011) provides an alternative distinction, namely between implicit and explicit concepts. While both kinds can contain statistical and causal information, explicit concepts are unique in their ability to represent syntactic information.

Words. How do explicit concepts fit into this puzzle? According to Fodor and Pylyshyn (2015), “that thoughts and sentences match up so nicely is part of why you can sometimes say what you think and vice versa” (p. 9). If Piccinini (2011) is right about the connection between explicit concepts and language, we might have a solution to apparent shortcomings within the existing models. Fodor and Pylyshyn (2015) have argued that concepts must compose to be able to form propositions. In response to the need for a mechanism of composition, Fodor and Pylyshyn have rejected the models that posit concepts with intensions or meaning, because they do not include a mechanism for how concepts can be combined. I agree that concepts need a mechanism of composition, but I do not believe the same mechanism needs to be responsible for storing and combining concepts. Furthermore, I think explicit concepts can connect the dots between the statistical/causal models and Fodor’s linguistic solution to the problem of compositionality. Although concepts are not words, words can serve as the mechanism by which concepts are mentally composed. Thus, the linguistic mechanism, which is typically associated with tasks such as speaking, is also responsible for composing associative and causal information. Items are represented as concepts in the mind and language gives us the ability to mentally combine these concepts. In this formulation, explicit concepts can compose while implicit concepts cannot. More precisely, the words that represent concepts explicitly in thought provide the means to combine concepts via syntactic rules.

Syntactic rules are great because they have the ability to represent an infinite number of thoughts. However, this strength is also a liability. How does a finite brain find useful things to say amongst the infinite number of options? Keil (1981) identified cognitive constraints as a method for discovering how people make sense of the world. He studied patterns in the way people in various age groups use predicates to span terms. For example, it makes sense to say a person can dream, but not that a rock can dream. So the predicate *dream* spans people, but not rocks. And it is not merely false to say that rocks dream, it seems impossible on a deeper level of reality. After a series of impressive experiments, Keil (1989) realized predicate-term relationships could be used to infer the ontological categories people are committed to. While syntactic rules give us a limitless mechanism to generate conceptual combinations, ontological constraints restrict the generative nature of syntax to help us identify meaningful combinations of words. Syntax need ontology like the fusion core of the sun needs gravity.

Developmental Readiness. What about how children go through developmental stages? How can we tell if students are developmentally ready for the abstract knowledge described in this study? Vygotsky and Piaget, both born in 1896, are perhaps the best known psychologists to propose developmental stages of the mind. I do not use the word obvious lightly, but it is obvious that children become more mentally sophisticated as they grow older. It is also obvious that children tend to develop physiologically as well. And it is a quite reasonable hypothesis to suggest that psychological development is causally related to physiological development, or at least follows a similar pattern. However, the correlation between the two may be spurious.

Fodor (1972) raised some important philosophical and methodological doubts about the underpinnings of Vygotskyan psychology. Vygotsky's developmental psychology is motivated by the Vygotsky blocks experiment where subjects are asked to classify blocks based on their attributes and the codes written on the blocks. Since the effectiveness of classifying blocks improves throughout childhood and peaks in adulthood, Vygotsky claimed that children pass through various stages of development as they grow older. The idea is that children struggle with simple block tests because they are not capable of forming the needed concepts. But as Fodor (1972) points out, children can classify faces and words, which are far more abstract and complicated than blocks. If children can perform sophisticated linguistic and facial recognition tasks, perhaps there is another variable involved which inhibits their block classification abilities.

Susan Carey's (1985) *Conceptual Change in Childhood* is an extensive rebuttal to Piaget's theory of development. After conducting multiple studies involving multiple age groups to find out how abstractions, biological concepts, and ontological categories develop in children, Carey rejects the existence of preoperational and concrete operational thinking. Rather than positing "changes in the types of reasoning" Carey suggests mental development is produced by "changes in the nature and organization" of knowledge (Carey, 1985, p. 135). Therefore, it is conceptual differences, not abstract reasoning abilities, that distinguish the judgements of children from adults. As for how to discover these conceptual differences, Carey recommends "that our deepest ontological commitments are to be analyzed in terms of our theories of the world" (Carey, 1985, p. 171). According to this view, conceptual change is the reorganization of ontological categories through differentiation, the splitting of categories, or coalescence, the merging of categories.

In a series of experiments, each involving dozens of preschool children from ages 3-5, Bullock, Gelman, and Baillargeon (1982) found that earlier studies relying on the verbal responses of children underestimated their cognitive abilities. The preschooler's explanations of events were often less advanced than their ability to make judgements or predictions. And as they got older and their explanations improved, there was not a corresponding improvement in judgements or predictions, except in cases where they knew more about the scenario. Although 3 year olds generally appeared to be less sophisticated than 4 and 5 year olds regarding causal mechanisms, there was evidence this resulted from a lack of knowledge and/or verbal abilities rather than a difference in reasoning style or the ability to evaluate causality. In a later study, Gelman and Markman (1987) demonstrated that preschool children, given perceptual information and knowledge about category membership, favor category membership when making inferences. Keil (1989) came to a related conclusion while studying predicate usage in children. He started by reviewing and contributing to evidence that young children tend to describe items using a variety of characteristic features, but as children get older, they shift towards using a more concise list of defining features. Keil (1989) discovered the characteristic to defining shift is more nuanced than previously thought. He found no evidence that it was caused by a sudden change in cognitive strategy as the child entered a new stage of development. Also, the data did not support the hypothesis that children are taught to focus on characteristic features rather than defining ones. In fact, adults themselves display the same tendency to focus on superficial characteristics when they lack theoretical knowledge about an item, especially without knowledge of an explanatory mechanism. Chi, Feltovich, and Glaser (1981) also demonstrated that novices tend to focus on surface features while experts tend focus on abstractions.

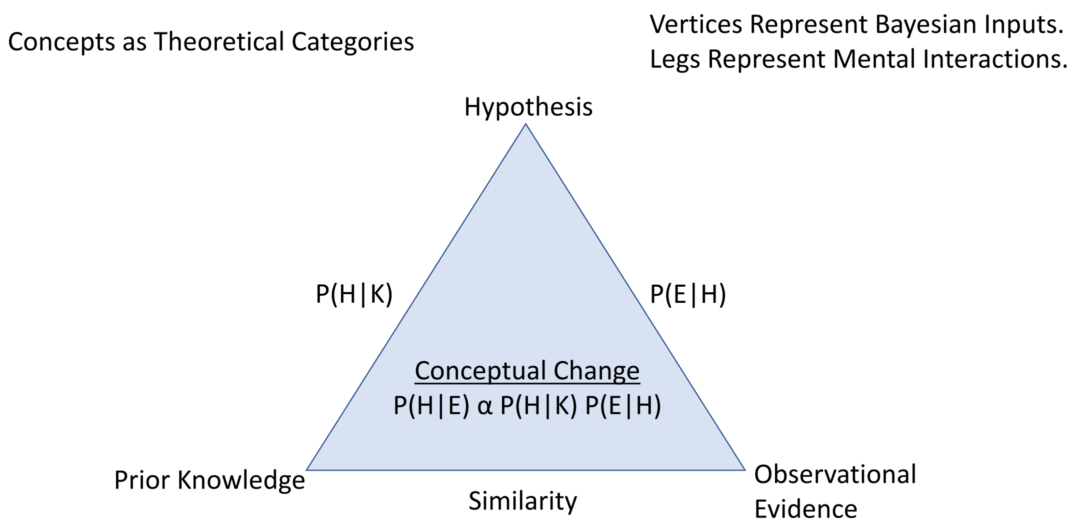
The point here is not to say there are no differences between experts and novices or adults and children, but rather to uncover less obvious ontological effects. As evidenced by patterns in predicate usage and development, children sometimes as young as preschool and often in kindergarten demonstrate ontological commitments in their claims about which predicates span a given term (Keil 1989). These ontological commitments imply that children are not solely focused on characteristic or surface features. Once again, in cases where child reasoning is different than adults, it is not their cognitive structure or strategies that are different; it is primarily their theoretical knowledge or lack thereof which drives these effects.

The Triangulation Model. What are concepts? We are now prepared to answer this question. A concept is a theory of a category. Categories are a set of items and concepts store the constraints used to delineate category membership. Specifically, concepts are theoretical categories based on exemplars, abstractions (e.g. prototypes), and causal relationships. Insofar as our prior knowledge is theoretical, it is also malleable. Theoretical knowledge can change when triangulated with observed evidence and hypothetical knowledge. A hypothesis can be proffered without any evidence. Observations can be checked for similarity without a hypothesis. But neither alone can change our mind. As for the mind itself, I am not precisely sure what it is. At the very least, it contains a likelihood calculator which requires three input parameters: theoretical knowledge, hypothetical knowledge, and observational evidence.

To unify the theories presented in this dissertation, I put forward the triangulation model, which is visualized in Figure 1.1 by connecting the three Bayesian inputs. At the base of the triangle, categorization by similarity occurs when prior knowledge is compared to observational evidence. Adding a hypothesis opens two additional interactions, giving us the ability to evaluate the probability of a hypothesis and the likelihood of the evidence given the hypothesis. When combined, these abilities provide a mechanism for conceptual change. Moreover, I believe overestimations of prior probability can explain synthetic conceptions (Vosniadou & Skopeliti, 2013) and ontological miscategorization (Reiner, Slotta, Chi, & Resnick, 2000). If the prior probability is overestimated and we are shown evidence that follows from the hypothesis, this provokes the acceptance of incoherence into our prior knowledge. From an educational standpoint, we often provide evidence to support a hypothesis. If the evidence is good, we assume the students will accept the hypothesis. However, from a cognitive standpoint, this might be backwards. Students could take the hypothesis as given and agree the evidence follows, without properly evaluating whether the hypothesis is consistent with their prior knowledge. This opens the door for mixing up categories that do not belong together.

This model also predicts that similarity does not produce conceptual change. In the absence of a hypothesis, new observations are judged based on similarity to prior knowledge. New evidence that is sufficiently similar is accepted. New evidence that is not similar is rejected. I suggest this is the mechanism for confirmation bias. When we include or exclude something from a category based on similarity, this does not change our underlying concept. Asking whether an item belongs to a category based on similarity is not a true test of the category. To falsify a concept, we need to use likelihood instead of likeness. Likelihood is a two-way street where confirmation and falsification are both possible outcomes.

Figure 1.1. The Triangulation Model



What exactly changes during conceptual change? Conceptual change occurs by splitting or merging an ontological category, changing the ontological category of a term, updating the predicates that span a category, or revising the causal relationships between or within categories (Carey, 1985; Keil, 1989; Rehder, 2003b). And new ontological hypotheses are evaluated in light of evidence and prior knowledge. As a possible mechanism, Griffiths & Tenenbaum (2009) have proposed that prior knowledge influences causal induction by "identifying which relationships are plausible and characterizing the functional form of those relationships" (p. 707).

The triangulation Model has compelling explanatory power. For example, consider the categorization task prompted by "Is this piece of fruit an apple?" Although this is logically a yes/no question, from a mental standpoint, it might prompt us to evaluate the likelihood that it is an apple. Because similarity often correlates with higher degrees of likelihood, it represents a reasoning shortcut. Our mind can speed up the reasoning process by leaving out the hypothesis apex of the triangulation model. In science, sometimes correlation meets our purpose and sometimes we need to work harder to find causation. In our mind, we are faced with an analogous choice between likeness and likelihood, where likeness is a shortcut for assuming likelihood. Scientists know causation takes precedence over correlation. Rehder (2006) has shown the mind knows this too.

Research Questions and Hypotheses

Purpose: To determine the effects of using Processing sketches to help students think Newtonian.

1. Will Force Concept Inventory results be the same for students who learn from coding physics simulations as compared to students from a traditional physics course?
2. How will student attitudes about learning physics change during a coding academy? Specifically, how will the simulations affect student recognition of real world connections and how will personal interest in physics be affected by the coding academy?

Hypothesis 1: Coding students will perform equally well on the FCI as compared to students from a traditional learning environment.

Hypothesis 2: Collectively, student personal interest in physics and ability to recognize real world connections will increase significantly during a coding academy.

Chapter 2: Literature Review

In this chapter, I will summarize evidence which supports the view that traditional physics instruction is ineffective at facilitating conceptual change. Because of this literature, we now know a great deal about the knowledge people obtain by observing motion in daily life and how this knowledge develops in childhood. Perhaps due to the predictability of physical interactions and the similarity in cognitive architecture between people, most of us end up knowing the same kinds of things about the way objects move. Fortunately for the job stability of physics teachers, the kinds of things people believe about motion are often in disagreement with Newton's laws.

One way to address the gap between common sense and Newtonian physics is through interactive simulations. In a two-year study involving 184 high school chemistry students, Pyatt and Sims (2011) found that students slightly prefer virtual to physical labs, but have a positive view of both, especially when the activity is inquiry-based. Zarcharia (2003) showed that teachers also have a favorable view of simulations, with or without being paired with laboratory experiments. Favorability is a good start, but I will also review literature regarding the effectiveness of simulations and I will provide suggestions for how physics teachers can improve upon existing methods.

Prior Novice Knowledge

With some areas of study, e.g. molecular orbital theory, our general experiences do not provide us with much relevant background knowledge. But our everyday lives are immersed in experiences with motion. This means novice physics students do not show up to class as blank slates; they arrive with knowledge about motion gained from personal experiences.

While analyzing results from the Demonstration, Observation and Explanation of Motion Test, Champagne, Klopfer, and Anderson (1980) found that "each student usually has a rich accumulation of interrelated ideas that constitute a personal system of common-sense beliefs about motion" (p. 1077). The same year, McCloskey, Caramazza, and Green (1980) discovered that university students often believe objects can move in curved paths in the absence of an external force. McCloskey and Kohl (1983) expanded the initial study with three additional experiments. They found the same curvilinear misconception in paper/pencil tasks, while viewing computer simulations, and while working directly with a moving physical object. Across a variety of contexts, in cases where Newton's first law predicts straight line motion, students believed the object would follow a curved path. McCloskey, Washburn, and Felch (1983) performed four more experiments to show the reverse could also be true, when students predict linear motion as Newton predicts curved motion. Their experiments looked at beliefs about the path of an object dropped from a moving reference frame. Newtonian physics has demonstrated how dropped objects with horizontal motion will follow a parabolic trajectory. But across a variety of contexts, up to 60% of students believed the dropped object would fall straight to the ground (McCloskey, Washburn, & Felch, 1983).

Halloun and Hestenes (1985a) developed and validated the Mechanics Diagnostic Test to study the common sense beliefs about motion among first time physics students at Arizona State University and a nearby high school. Pre-test results were generally low, especially among high school students. Test gains after taking a conventional physics course were generally small and not dependent on the instructor. These results demonstrate how difficult it can be "to facilitate a transformation in the student's mode of thinking from his initial common sense knowledge state to the final Newtonian knowledge state of a physicist" (Halloun & Hestenes, 1985b, p. 1048). In a related study, Halloun and Hestenes (1985b) interviewed 22 students one month after they took the Mechanics Diagnostic Test. These interviews were used to develop a taxonomy of common sense beliefs about motion, organized into two categories, "principles of motion" and "influences on motion" (Halloun & Hestenes, 1985b, p. 1063). For example, students often believe motion always has a cause and many further contend this cause is some kind of internal force. Overall, Halloun and Hestenes (1985b, p. 1056) recommend looking beyond the term "misconception" by viewing student prior knowledge as a set of "alternative hypotheses to be evaluated by scientific procedures."

Tao and Gunstone (1999) developed the Force and Motion Microworld, a set of four computer simulations designed to address the alternative conceptions of 10th grade students in Australia. Despite a relatively small sample size of 12 students, they identified some important ramifications for conceptual change. First, regardless of engagement with conflicting evidence, students did not always experience conceptual change, especially without reflecting on the discrepancy. Second, despite the eventual acceptance and usage of scientific conceptions among most participants, students continued to selectively apply their alternative conceptions in a context dependent way.

Virtual Expert Knowledge

In a study of 250 non-biology major college students, Windschitl and Andre (1998) used two different cardiovascular simulations to show the difference between constructivist and objectivist learning. The constructivist simulation emphasized exploration while the objectivist version emphasized confirmation. Epistemological survey results were compared to posttest results and while the constructivist group overall performed slightly better than the objectivist group, students with more advanced epistemological beliefs learned particularly well from the exploratory simulation (Windschitl & Andre, 1998). Students who reported being motivated by factors unrelated to understanding the content tended to become frustrated with the constructivist simulation and performed better with confirmation learning. Steinberg (2000) found similar mixed results while comparing an interactive simulation of air resistance with a traditional paper and pencil tutorial. On the exam, he found no significant difference between the two groups. However, a greater range of engagement was observed within the interactive group. Steinberg (2000) reported that some students simply wrote down the answers provided by the computer without much investigation while other students delved deeper into the simulation and tried more complex experiments. The behaviors identified by Steinberg (2000) seem to correspond well to the objectivist and constructivist attitudes identified by Windschitl and Andre (1998).

The Physics Education Technology project was created by Carl Wieman at the University of Colorado Boulder (PhET, 2017). The PhET website has well over 100 simulations spanning physics, chemistry, biology, earth science, and math. In a 15-week study involving 231 students, Finkelstein, Adams, Keller, Kohl, Perkins, Podolefsky, Reid, and LeMaster (2005) showed PhET circuit simulations to be more effective than physical circuit labs at promoting conceptual understanding. Moreover, in a follow up task, the simulation students outperformed the physical circuit students at actually constructing real circuits. Zacharia (2007) had similar findings with 88 undergraduate students using electric circuit simulations. In his study, students who learned from virtual and real circuits demonstrated better conceptual understanding than students who studied real circuits alone. When virtual circuits and real circuits were tested separately, students using the simulations outperformed the physical circuits group. In a subsequent study with 62 undergraduate students, Zacharia, Olympiou, and Papaevripidou (2008) showed virtual and physical manipulatives to be equally effective during inquiry labs involving heat and temperature. They also found evidence of virtual experiences enhancing physical labs based on a comparison of pre- and post-test results.

Wieman, Adams, and Perkins (2008) suggest the following design strategies for simulations: an environment that is interactive, challenging, and dynamic without being overwhelming. In terms of teaching strategies, Wieman, Adams, Loeblein, and Perkins (2010) recommend addressing prior knowledge, making connections to real-world experiences, and to encourage exploration. After reviewing 79 studies, Scalise, Timms, Moorjani, Clark, Holtermann, and Irvin (2011) identified best practices for educational simulations. In the inquiry cluster, they emphasize scientific questions, evidence, investigation, explanations, and justification.

From here on, try to keep Wieman's PhET project in mind. We do not have to start from scratch when good ideas are already on the table.

Processing Motion

To Turtle. In 1964, after spending five years working with Piaget in Geneva, Papert went to MIT and became the lead developer for the LOGO computer language (Papert, 1980). LOGO was invented to program artificial turtles. Papert described the turtle as “a computer-controlled cybernetic animal” (Papert, 1980, p. 11). Some of these turtles were physical robots, but most were displayed virtually on computer screens. LOGO is primarily used as an educational tool for teaching students how to represent their ideas with code. By typing the PD command, which stands for PENDOWN, and a motion command like FD x100, which commands the turtle to walk FORWARD 100 pixels, users can draw geometric shapes on the screen. With the right procedure, you can draw virtually any shape or pattern. After LOGO’s inception, computers became faster, the syntax became richer, and people got better at coding. By 1980, there was a textbook describing turtle vector graphics, 3D shapes, and even how to represent the curved space of general relativity (Abelson & diSessa, 1980).

Turtles are not restricted to geometry; they can also do dynamics. Kolodiy (1988) showed how to teach projectile motion with LOGO turtles. Then, with Papert as his doctoral advisor, Resnick (1994) produced StarLogo, a simulation language that can draw thousands of turtles on the same screen to represent massively parallel and highly decentralized systems of behavior. He called StarLogo projects “microworlds” which are “specially designed to highlight (and make accessible) particular concepts and particular ways of thinking” (Resnick, 1994, p. 50). Resnick has produced microworlds to help understand systems like ant colonies, traffic jams, and forest fires. In these situations, large collections of simple localized interactions can give rise to complex patterns. These microworlds demonstrate how large-scale emergent phenomena can occur in the absence of a centralized cause.

Processing. Processing was developed at MIT by Casey Reas and Ben Fry. The first version was released in 2001. As a language designed to generate and modify images, Processing gave new media artists greater control over their work (Reas & Fry, 2006). With an emphasis on imagery and a straightforward syntax, beginners can see quick results while using Processing. Reas & Fry (2014) view software as a unique method of artistic expression because it can "produce dynamic forms, process gestures, define behavior, simulate natural systems, and integrate other media including sound, image, and text" (p. 1). The present study needed a programming language that is simple enough to learn quickly and powerful enough to simulate Newtonian mechanics. Processing was up to the task.

Dan Shiffman is an associative arts professor at NYU, an author, a Director of the Processing Foundation, and the star of Coding Train on YouTube. His book *The Nature of Code* is the definitive text on using Processing for simulations (Shiffman, 2012). In the preface, Shiffman considers whether it is a science book or an art book. He claims it is neither. I believe it is both. Many of the programs Shiffman writes are microworlds in the Resnick (1994) sense of the word. Although these simulations are not representative of full blown reality, they visually depict natural behaviors with clarity and precision. Shiffman uses these exercises primarily to teach coding. By emphasizing Newtonian thought, this approach can also be used to teach physics.

Consider projectile motion and imagine tossing a ball from a roof. Traditional instruction involves drawing diagrams and demonstrating how trigonometry can be used to solve the problem. Students might very well agree the diagrams and trigonometry are useful, but this does not address underlying misconceptions. If students are asked to implement a projectile motion model using code, it offers a new view of the problem. Coding for a computer screen is inherently 2-dimensional. If we draw a circle to represent the ball, we must use a line of code like this:

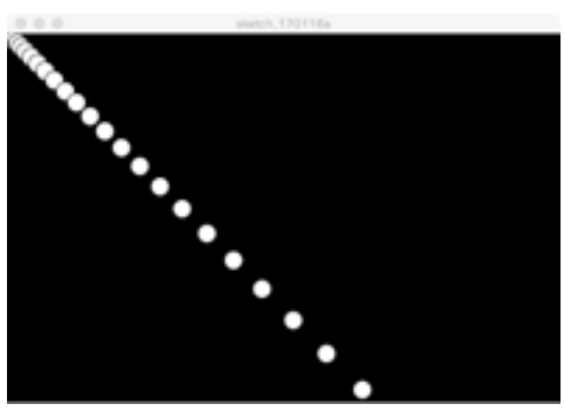
```
ellipse(x, y, size, size);
```

where size represents the pixel width and height of the ball. The x variable represents the horizontal position and y represents the vertical position. The code requires us to think about the horizontal and vertical motions independently. If students believe the horizontal and vertical motions are both accelerated by a force, they could test this model with the following code:

```
x += ax*t*t;
y += ay*t*t;
```

where a represents acceleration and t represents time. Surprisingly, this will result in a straight diagonal path, which does not match experimental evidence.

Figure 2.1. Diagonal Path



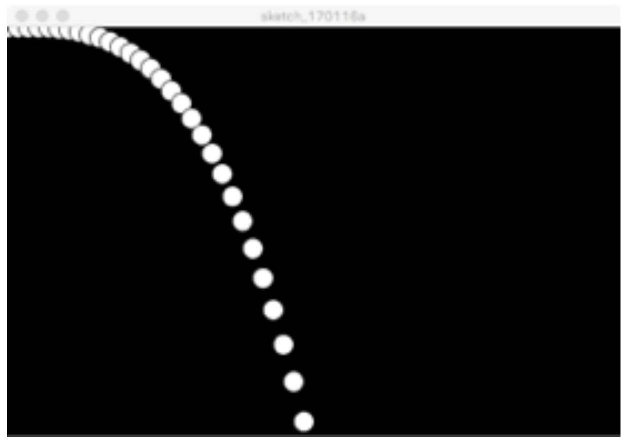
On the other hand, if students believe the horizontal motion is constant and vertical motion is accelerated by a downward force, they could test the model with:

x += vx*t;

y += 0.5*a*t*t;

where vx represents horizontal velocity. With this model, we get the expected parabola.

Figure 2.2. Parabolic Path



Computer simulations can therefore provide students with a way to test their ideas by visualizing mathematical models.

Before moving on, I would like to explain a full Processing sketch, to offer a better sense of the syntax. The code is on the left and my comments follow // on the right.

Figure 2.3. Example of Processing Sketch

```
float d = 1;          //the first four lines all declare variables
float a = 0;          //float variables can store decimals
float j = 1;          //each variable is set to its initial value
float k = PI/1000;

void setup() {        //the setup function runs once
  size(800, 800);           //sets the pixel size of the sketch window
  background(0);            //sets the background color, 0 is black
}
```

```
void draw() {                          //draw function runs repeatedly
  translate(width/2, height/2);        //moves origin to center
  noStroke();                                   //turns off outline of shapes
  d += j;                                               //addition equation
  if(d > width/2 || d < 0){                     //if statement, || means OR
    j *= -1;                                    //multiply equation
  }
  a += k;
  fill(255, 0, 255);                            //sets shape color to pink
  ellipse(d*cos(a), d*sin(a), 3, 3);   //ellipses with 3-pixel diameter
  ellipse(d*sin(a), d*cos(a), 3, 3);   //trig controls emergent shape
}
```

The preceding sketch will draw something that looks like an abstract flower pattern. If you want to run the code, visit www.processing.org to download the freeware. I used Processing 3 for this study, but a newer version of Processing called P5.js is also available free of charge.

Teaching Physics with VPython. Before summarizing this chapter, I will review two studies which used the VPython computer language. These VPython studies were both in different contexts and had different designs, but they share an important similarity with the present study. They involve using code to teach physics.

At a university in South Africa, Buffler, Pillay, Lubben, and Fearick (2008) conducted a study with 51 students. Thirty-three of the students reported no previous experience with coding. Students were taught how to control a simulated spaceship with VPython commands. The students with no coding experience, sometimes with peer help, were able to complete the task after one afternoon of practice. The results showed that students can quickly learn coding skills in the context of a physics lesson.

Also working with university students and VPython, researchers at Georgia Tech studied the results of 1357 students using code to complete homework problems during a one-semester physics course (Caballero, Kohlmyer, & Schatz, 2012). One of the 14

Georgia Tech assignments was similar to the Buffler et al. (2008) spacecraft task. They found no statistical difference between traditional homework question scores and coding scores, so students were able to express their physics knowledge with and without the coding environment. They also confirmed that students with little or no prior coding experience could quickly learn how to work with VPython to solve physics problems.

The present study took place over two weeks, representing an intermediate between the two VPython studies which lasted one afternoon and one semester. My students were younger and we used Processing instead of VPython, but these two VPython studies showed that inexperienced students can learn to code quickly in the context of a physics class. Given the evidence I have reviewed regarding young children's ability to think abstractly, it is reasonable to suggest teenagers would be able to complete abstract coding tasks like their university counterparts.

Summary of Literature Review

As a physics teacher, I want to be able to teach my students Newtonian mechanics. But mechanics is about everyday phenomena and students show up with theories about how motion works. If this prior knowledge was generally consistent with Newton's view, it would make my job easier, but it is not, so things are not so simple. Simulations have shown promise as a means to promote conceptual change. Supported by the theoretical framework from Chapter 1, I aim to improve the educational benefits of simulations by asking students to write their own simulation code. Coding represents a technique for bringing together hypothetical knowledge, prior knowledge, and observational evidence. Shiffman has shown that physics helps when you are learning to code. Papert and Resnick have shown it also works the other way around; code can help you learn physics. In Chapter 3, I will take the Papert-Resnick LOGO method, move it to the more powerful Fry-Reas-Shiffman Processing environment, and produce an educational sequence that is guided by PhET design principles and the triangulation model.

Chapter 3: Methodology

In his 1984 book *Vehicles*, Braitenberg imagines a variety of machines that are programmed to move and respond to their environment. In successive chapters, the vehicles acquire more sophisticated sensors and more complex behaviors. To the reader who follows along with Braitenberg's recipe for inventing these machines, they are easy to understand. But Braitenberg imagines trying to discern the inner workings of such machines by analyzing their behavior. He claims this would be very difficult since different designs can produce the same effects. Along these lines, Braitenberg (1984) states "when we analyze a mechanism, we tend to overestimate its complexity. In the uphill process of analysis, a given degree of complexity offers more resistance to the workings of our mind than it would if we encountered it downhill, in the process of invention" (p. 20). I believe coding can help students learn about motion by transferring analytical tasks to an inventive space. Pre-made simulations give students a way to analyze laws of motion. Coding provides a way to invent laws of motion.

Educational Methodology

The triangulation model is a representation of Bayesian learning and similarity-based reasoning. It predicts conceptual change occurs primarily by evaluating hypothetical knowledge in light of evidence and prior knowledge. This evaluation has two steps: determining the likelihood of the hypothesis given prior knowledge, $P(H|K)$, and the likelihood of the evidence given the hypothesis, $P(E|H)$. It is important to note that $P(H|E)$ has a higher standard than $P(E|H)$ while $P(H|K)$ determines how much higher. Suppose I leave a sandwich on the table and when I return the sandwich is gone and a dog is sitting in my chair. A reasonable hypothesis would be the dog ate my sandwich.

However, it is possible someone else ate the sandwich or that it has been misplaced but not eaten. So, P(H|E) is not unity. But if the dog did eat my sandwich, the sandwich would in fact be gone, so P(E|H) is noticeably higher than P(H|E). If I ignore P(H|K), I am going to expose myself to the fallacy that P(H|E) can be judged solely based on P(E|H), which will cause me to overestimate the likelihood of the hypothesis. This is why prior knowledge plays such a big role in reasoning.

Teaching with pre-made simulations can provide mixed results because it is hard to guarantee students will take advantage of the interactivity to make connections to prior knowledge. It is possible to watch a simulation and take it at face value without considering what you already know. But when students are writing their own code, the process is inherently tied to interactivity and prior knowledge. Developing code requires careful thought and constant adjustment often while experimenting through trial and error.

On the following page, there is a guideline provided to students which explains how to write a simulation. These steps were used to teach students how to think about simulations, but so long as they were reasoning based on prior knowledge and evidence, the steps were not strongly enforced. I did not want something that looks like a worksheet to inhibit exploration.

During each coding academy, students were taught how to code a vector-based physics engine. Then, students were asked to apply their physics engine to complete challenging projects. Hake (1998) used a data set with 6000 students to show interactive-engagement courses produce twice the conceptual gain of traditional courses as measured

Figure 3.1. Steps for designing a simulation model

1. What phenomenon are you going to simulate? Write a hypothesis about how it works.	
2. Translate your hypothesis into a math equation and define all variables.	
3. Insert your equation into a simulation and test it. Does the simulated evidence match your hypothesis from Step 1?	
YES – Go to Step 4	NO – Return to Step 2
4. Does the hypothesis match what you knew about motion before the simulation?	
YES – Go to Step 6	NO – Go to Step 5
5. Should you revise what you know or revise the hypothesis?	
Knowledge – Go to Step 6	Hypothesis – Return to Step 1
6. Show your simulation to other people and ask if it is visually overwhelming.	
YES – Redesign, then go to Step 7	No – Go to Step 7
7. Discuss your simulation with other people. What real world situations demonstrate this kind of motion? Be specific and list your examples.	

by the force concept inventory. Given the highly interactive nature of the processing motion strategy, it is reasonable to expect some strong FCI scores. However, most of the Hake (1998) samples resulted from a semester of instruction and my coding academies lasted only two weeks. This put us at an ambitious time scale for seeing conceptual change.

Experimental Methodology

Sample. During summer 2017, thirty-two students participated in coding academies which were studied as part of this dissertation project. The coding academies were held at two different inner-ring suburbs of a large city in Missouri. None of the academies reached full capacity, so all students who applied to participate were invited to join.

I reached out to eight school districts and made extra efforts to sign up schools with higher poverty rates, identified by greater than 50% free or reduced lunch. Since the coding academies were free to the schools and to the participants, it made sense to invite students who are less likely to have money for summer camps or academies. The schools were recruited by emailing and calling district administrators or teachers and by following up with visits to their site. I recruited participants by visiting their classroom and sharing paper or electronic promotional materials. When possible, I brought a current student with me to provide a learner's perspective of coding with Processing. The sign-up form contains the questions listed in Table 3.1. Hudson and McIntire (1977) showed that mathematical ability correlates weakly with performance in physics class, so I will report the student's background in math with my results.

Research Questions.

1. Will Force Concept Inventory results be the same for students who learn from coding physics simulations as compared to students from a traditional physics course?
2. How will student attitudes about learning physics change during a coding academy? Specifically, how will the simulations affect student recognition of real world connections and how will personal interest in physics be affected by the coding academy?

Table 3.1. Academy Sign Up Form

1	What is your name?
2	What is your email address?
3	How old will you be on June 15, 2017?
4	What is your parent/guardian's name?
5	What is your parent's email?
6	What school do you attend?
7	Have you received credit for taking high school physics?
8	Have you received credit for taking high school algebra?
9	Have you received credit for taking high school geometry?
10	Have you received credit for taking college level physics (including AP)?
11	Are you fluent in any computer programming languages?
12	Which academy location do you prefer?
13	The academy will be at a school and supervised by a certified teacher, so participation will be as safe as attending summer school. In case of an unexpected accident, please provide emergency contact information here.

Measuring Concepts. The Mechanics Diagnostic Test (MDT) was designed to measure the conceptual understanding of physics students (Halloun & Hestenes, 1985a/b). The diagnostic test helped confirm how student conceptions are quite different from expert physicists. MDT was also the precursor to the Force Concept Inventory (Hestenes, Wells, & Swackhamer, 1995) and the Mechanics Baseline Test (Hestenes & Wells, 1995). The FCI contains 29 items with strong common sense distractors. After administering the FCI to more than 2000 students, Hestenes, Wells, and Swackhamer (1995) suggest "pretest scores are so uniformly low for beginning physics students that further pretests are really unnecessary, except to convince diehard doubters or to check out the conceptual level of a new population" (p. 150). Given the risk of a pretest interaction which would compromise the validity of the FCI results, I only administered a posttest at the end of the coding academy. Henderson (2002) found no significant influence of pretest on posttest when they were administered a semester apart, but with the much shorter study duration, it is more likely that I would have seen a difference. Henderson (2002) also demonstrated that students generally take the FCI seriously even when it is ungraded. He found a 2.8% drop in FCI scores in the case of being ungraded. About 1.9% of this reduction is identifiable in students who either refuse to take the test or leave the test mostly blank. The coding academies were not for credit, which could have reduced scores a bit, but 2.8% seems like a fair price to pay.

The Mechanics Baseline Test (MBT) is a 26-item test designed for more advanced students than the FCI. When Hestenes and Wells (1992) plotted FCI results vs. MBT results, they noticed data clustering which suggested students generally need to score 60% on the FCI to hit 60% on the MBT and the same thing occurs at 80%. They call the 60% mark a "conceptual threshold" and 80% a "mastery threshold" (Hestenes & Wells, 1992, 161). As a historical note, the results of these tests inspired a search for better ways to teach physics. Physics modeling is arguably the most important of the new methods of instruction and Hestenes (1996) provides an extensive summary.

After performing a factor analysis involving 145 high school students and 750 university students, Huffman and Heller (1995) criticized the FCI for not factoring into the categories suggested by the authors of the test, namely kinematics, first law, second law, third law, superposition principle, and kinds of force. In the high school sample, the third law dimension items correlated well and some of the kinds of force items were grouped by the analysis. However, while studying a similar age group in Finland, Savinainen and Scott (2002) found students to be inconsistent with 3rd law problems. In their study, students were twice as likely to get item 16 correct as item 15, but both belong to the 3rd law dimension. Since item 15 involves acceleration and item 16 is a constant velocity problem, Savinainen and Scott (2002) suggest that students might have trouble generalizing 3rd law interactions. This result implies that even the limited FCI groupings found by factor analysis may not hold in other samples. With 647 students, Stewart, Griffin, and Stewart (2005) determined that context sensitivity is not responsible for the variations between student responses. In Huffman and Heller's (1995) university sample, four items grouped together, three of which were from the kinds of force category. And the 3rd law category did not factor together in their university sample. Because most of the 29 items do not form statistical categories, Heller and Huffman (1995) propose that FCI items might be measuring subconceptual knowledge, but not a coherent force concept. In response, Hestenes and Halloun (1995) advise analyzing the

FCI as a whole when given to students. They also note the six conceptual dimensions were based on the expert physicist's perspective and not the non-Newtonian perspective. And since the distractors are so powerful, they tend to reduce the coherence of student responses. This view is supported by Savinainen and Viiri (2007) who interviewed 49 high school students after taking an FCI posttest. They discovered high conceptual coherence among students reaching mastery level (>80%) on the FCI. In other words, the categories emerge as expertise increases. Next, I will consider the Huffman and Heller (1995) results in terms of the triangulation model.

We are accustomed to categorizing objects based on the similarity of their superficial features. But finding similarity between causal relationships is more difficult. For example, it is not obvious the force pulling us towards the ground shares the same cause as the force keeping the earth in orbit around the sun. Orbits and projectiles are superficially different, but they share the same cause. I suggest the lack of theoretical knowledge to make this connection only makes the categories look incoherent. Although the FCI items form distinct factors if you possess the theoretical knowledge needed to see the causal similarity, the factors will not arise in samples where students lack the relevant theoretical knowledge. More broadly, this mechanism could apply to the problem of transfer. Transfer is difficult because it requires us to find abstract or causal similarity between two superficially different phenomena. In my view, if we more often use similarity as a shortcut for causal reasoning, it would seem peculiar to look for similarity between two causal relationships that look nothing alike.

Measuring Attitudes. The second research question is aimed at determining how student attitudes are affected by learning to code Newtonian simulations. Since they worked in a virtual environment, we might expect students to not recognize the connections to real-world applications. However, by explicitly asking participants to draw information from their prior knowledge, this opens a path for students to make such connections.

The Views About Sciences Survey, or VASS, contains 30 items organized into six dimensions (Halloun & Hestenes, 1998). The dimensions were verified by analyzing the responses of physics teachers and professors. According to Halloun and Hestenes (1998), VASS can distinguish between “naive realists,” who are 22 times more likely to be passive learners, and “scientific realists,” who “believe the physical world cannot be known directly through sense perceptions, but only indirectly through theoretical constructions” (p. 572). However, because VASS correlates well with the FCI, administering both instruments to the same sample may turn out to be an exercise in correlating results that are already known to correlate. FCI gains have also been correlated to the Lawson Classroom Test of Scientific Reasoning (Coletta & Phillips, 2004). The Lawson test does not directly measure attitudes, but I think it could be used to infer ontological beliefs. Regardless, I decided to look for other instruments to pair with the FCI.

The Maryland Physics Expectations survey (MPEX) is a 34-item instrument and like VASS, is also organized into six dimensions (Redish, Saul, & Steinberg, 1998). Unlike VASS, MPEX contains a reality link cluster which measures student ability to make physics connections to events outside the learning environment. When Redish, Saul, and Steinberg (1998) analyzed MPEX data from over 1500 students at six different universities, they found significant decreases in reality link scores after physics instruction. In other words, taking a physics course can make students less likely to acknowledge affiliations between physics concepts and the real world. Elby (2000) used the MPEX and the Epistemological Beliefs Assessment for Physical Science (EBAPS) to identify instructional practices that improve expectations and epistemological beliefs of physics students. He found significantly more favorable results on the MPEX and EBAPS by embedding epistemology into class and homework, grading based on effort of explanations, and reducing textbook use and content coverage. MPEX could certainly help answer my research questions about student attitudes, but there is a newer instrument which is better suited.

The Colorado Learning Attitudes about Science Survey (CLASS) was administered to over 5000 people and validated with interviews, reduced basis factor analysis, and a reliability study involving hundreds of college physics students (Adams, Perkins, Podolefsky, Dubson, Findelstein, & Wieman, 2006). It contains 42 items and 8 categories including real world connections, personal interest, sense making/effort, conceptual connections, applied conceptual understanding, problem solving general, problem solving confidence, and problem solving sophistication. The real world connections and personal interest categories match the purpose of the present study, so I chose to administer CLASS to participants in the Processing Motion coding academies. Some of the items are not scored. Item 31 is used to determine if people are reading the statements. Items 7 and 41 did not pass the reliability test. Items 7, 9, and 33 do not fit into any of the categories. Because these unscored items were included during validation and do not represent a major burden to participants, I left them in the survey.

CLASS has been adapted for use in biology (Semsar, Knight, Birol & Smith, 2011) and experimental physics (Zwickl, Hirokawa, Finkelstein, & Lewandowski, 2014). The experimental physics version (E-CLASS) passes a variety of validity tests, but does not demonstrate strong factors like CLASS (Wilcox & Lewandowski, 2016). Virtual simulations can be analogous to experiments, so I considered using E-CLASS for this study, but decided to stick with CLASS since the real world connections factor is important to my second research question. Next, I will go over a few CLASS results that are relevant to my project.

Brewe, Kramer, and O'Brian (2009) found positive attitude shifts while using modeling instruction with 45 introductory physics students. Milner-Bolotin, Antimirova, Noack, and Petrov (2011) showed that taking physics in 12^{th} grade correlates to more favorable CLASS results in absolute terms during college courses, but students who did not take physics 12 tend to have greater relative gains. This is important to note, since none of the participants in the processing motion study have taken 12^{th} grade physics. Also, in an experiment involving 176 high school students, Marušić and Sliško (2012) studied attitudinal change differences between groups learning from a reading/presenting/questioning method and an experimenting/discussion method. The reading group saw overall 5.8% positive gains on CLASS with no statistically significant negative shifts. The experiment group had five statistically significant positive categories, no negative shifts, and an overall 25.6% gain. Lastly, where Elby (2000) found success by explicitly including epistemology into his lessons, Lindsey, Hsu, Sadaghiani, Taylor, and Cummings (2012) measured significant attitudinal shifts with inquiry lessons focused on conceptual development. This means lessons which implicitly address epistemological issues in science can change student attitudes. That is good to know because my educational methodology is more focused on coding than epistemology. But there is one epistemological issue that stands out. Hammer (1994) found that some students do not realize that physical laws are supposed to be coherent. Thus, the lack of coherence in physics reasoning cited throughout this paper could be more of an epistemological misunderstanding than a conceptual one.

Limitations. Regarding this research project, there are a few concerns related to validity. Because the study includes self-selected volunteers, it is possible the sample is not fully representative of the broader population of students. To the extent that my sample is not representative, this could impact the external validity of the results. I also did not form a new control group. However, people are uniformly ineffective at solving Newtonian physics problems, unless they have received strong training. So long as the participants have not received similar training before joining the study, this means extraneous variables pose little threat to internal validity. Given the ethical implications of withholding training from a control group, with little to be gained on the validity side and the possible risk of compensatory rivalry, it makes sense to use pre-existing data as a control.

Where possible, I tried to isolate the programming effect as much as possible. For example, if I had used instructional methods unrelated to programming during the experiment, it could be these methods that affected the FCI results instead of the programming method. Traditional physics teaching involves lecturing, labs, textbook, and homework problems. In this study, large group lectures were primarily used for housekeeping and introducing programming concepts. Neither of those are assessed on the Force Concept Inventory, so lectures should not have directly impact the results. Instead, the Cain and Laird (2011) approach of Frequent Small Group Purposeful Talks were used and this approach is harder to separate from the results. I had frequent conversations with individuals and small groups during the coding sessions. These were in the context of programming, but I did receive questions about physics. When possible, I answered by directing students to resource materials or to use the computer program itself to find the answer. The resource materials can be viewed in the Appendix.

During the academies, there was no physical lab equipment, so this method will not impact the results. And to the best of my knowledge, students did not spontaneously perform any lab experiments with ambient materials.

We did not use a physics textbook during the academy, but some supporting materials were provided regarding the laws of Newtonian physics and coding syntax. Again, these can be found in the Appendix. The reference materials played an integral role in the process of learning to code Newtonian simulations. However, materials like this have been provided to countless people, many of whom did not become proficient at Newtonian physics, so it is unlikely that simply possessing a reference sheet increases your proficiency with the content.

Of the four traditional methods, writing code is the most similar to pencil and paper problem solving. The methodology is superficially different, but the mathematical reasoning is essentially the same. However, with traditional problem solving, it is common to focus on one instant in time. For example, what was the velocity at t = 4.3

seconds? With coding, we are compelled to figure out how each iteration through the code will affect the results of the equations. We cannot merely focus on $t = 4.3$ s. Code requires us to think about how the equations will interact for the duration of the program. This gives coding an advantage over paper and pencil. Regardless, coding a physics simulation requires problem solving.

The experimenter effect arguably poses the greatest threat to validity. If the effects are unique to my personal style of instruction, the results may not generalize. The best approach would be initiate a second study where other teachers try the same method.

Chapter 4: Results

Pilot Academy

Approximately ten students wanted to sign up for a coding academy but were unavailable for the scheduled dates. Four of these students were available the week before the first academy. The computer lab was setup early at the first school and we were within the IRB approved dates, so I decided to hold a one week pilot academy prior to the full two week academies. The pilot group included four girls with a mean age of 13.5 years. All four pilot students took a CLASS pre-survey and received the first half of the lessons. Two of the students followed up after the full academy to work on the missed lessons. This was done primarily as an independent study with email communication, but we met once again at the school so the students could take the CLASS post-survey. The Force Concept Inventory was not administered to any of the pilot students.

The pilot academy provided early observations to help implement the coding lessons for the full academies. The initial plan was to start by teaching the students to code one-dimensional physics equations and then build up to vector based equations. However, the coding students generally did not recognize the standard forms, so referring to them did not give us a good starting point. It was easier to skip straight to the vector equations. This approach contrasts with my prior 11 years of experience teaching high school physics. On paper, my students often struggle with vector equations. In this new context, with the same equations represented as code, students had no trouble using the first coordinate to manipulate x-values and the second coordinate for y-values. For the full academies, I did not introduce the textbook version of physics equations. As soon as the students started moving objects on the screen, they were taught about vectors.

While writing simulation equations, I observed that students have a strong bias towards addition. Their attempts to write motion equations almost universally start with addition. Thus, the physics engine involving two equations position.add(velocity) and velocity.add(acceleration) were intuitive to the students. In the context of coding, I also found students were better able to visualize and interpret adding a negative value as compared to my experiences in the classroom. On paper, students seem more comfortable with subtraction than adding a negative, despite these operations being mathematically identical. However, the coding students had no trouble adding negative values and predicting the reversing effect in their code.

Before moving to the full academies and the quantitative data, I want to describe two more important observations from the pilot. First, it helps to ask students to regularly start new programs or sketches. Otherwise, they tend to keep adding more and more code to an existing sketch and it loses coherence and purpose. Second, it helps to ask students to type all their code, even when they are borrowing an idea from the teacher, other students, or the internet. Students can learn a great deal from interacting with the code of others. But if they simply copy and paste code, it does little to help them learn the syntax. During the full academies, when I provided code, it was always given on paper and students were asked to type it and make changes or additions to meet a goal.

Full Academies

Both academy locations were inner ring suburban high schools of a large city in Missouri. The pilot and first academy took place at School 1. The second academy took place at School 2. Table 4.1 summarizes publicly available data about both high schools (MO DESE, 2017). Table 4.2 summarizes data about the student samples participating in the summer academies.

Table 4.1. School Data

2016/17	Enrollment	Black	White	Other	Free/Red. Lunch	4 Yr Grad Rate	ACT	Disciplinary Action
School 1	302	32.8 %	54.6 %	12.6%	51.5%	91.4%	20.4	2.3%
School 2	841	88.6%	8.2%	3.2%	64.3%	86.0%	17.3	8.6%

Table 4.2. Academy Data

Summer 2017	Enrollment	Black	White	Other	Male	Female
Pilot at School 1	4	0 %	75 %	25%	0%	100%
Academy at School 1	13	31%	69%	0%	69%	31%
Academy at School 2	15	53%	20%	27%	80%	20%

The full academies were both two weeks long. We met from 10:00 AM – 12:30 PM on weekdays. The attendance rate for the first academy was 89.2% and the second academy attendance rate was 84.7%.

Each academy session started with a short full group discussion. On the first day, we discussed the basics of the academy. This included reviewing computer hardware/software, the schedule, and expectations. After the first day, each session began with a full group discussion that was question driven. The list of questions is in Table 4.3. Discussion questions were introduced in the sequence listed, but each day, I brought up previous questions as needed to remind students what we discussed on previous days. Initially, I did not plan to discuss physics with the full group because I was concerned about compromising the focus on coding. But during the pilot, students were using physics terminology inconsistently, so I started asking the full group questions about the meanings of common physics terms. The full-group discussions were primarily used to standardize the way terminology was used by students during the academies. Discussions were also used to explore the analogy between pixels and position and the relationship between the computer frame rate and time. Academy project descriptions can be found in the Appendix.

In addition to myself as instructor, there was a tech aide present at each session. I hired two former students, each a recent high school graduate, at a rate of $10 per hour to help with the academies. The tech aides helped resolve computer hardware/software malfunctions and generally helped the students troubleshoot technology problems. I focused on coding instruction, but there was some overlap. Tech aides also answered coding questions and I helped with technology problems from time to time. I was not compensated for teaching the summer academies, except for being able to collect data.

Table 4.3. Discussion Questions

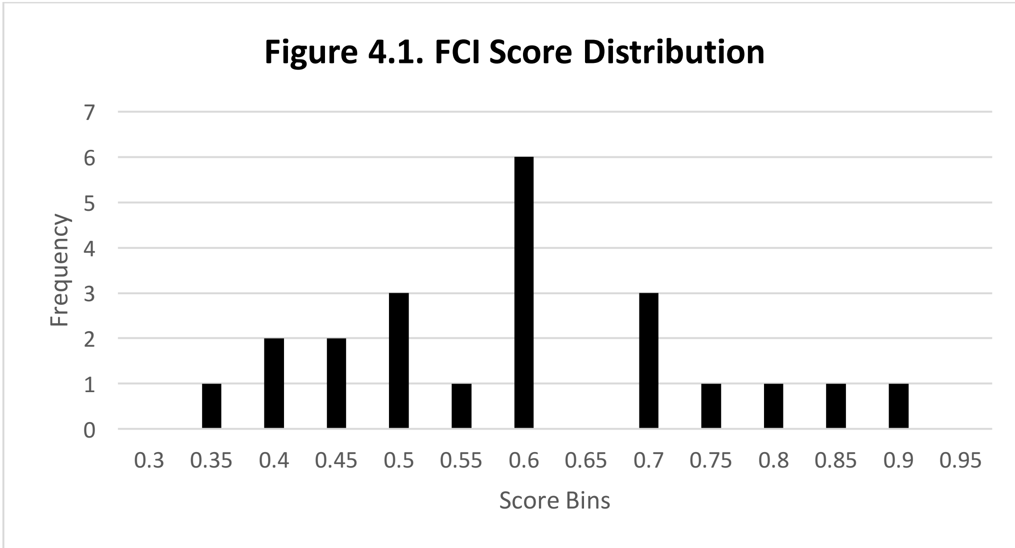
Question 1	What is coding?
Question 2	Why would we want to represent the physical world on computers?
Question 3	What is position?
Question 4	What are pixels?
Question 5	What is time?
Question 6	What is velocity?
Question 7	What is acceleration?
Question 8	What is a vector?
Question 9	What are forces?

Question 10	Is there a minimum distance or time in the real world?
Question 11	Is there a minimum distance or time on the computer screen?

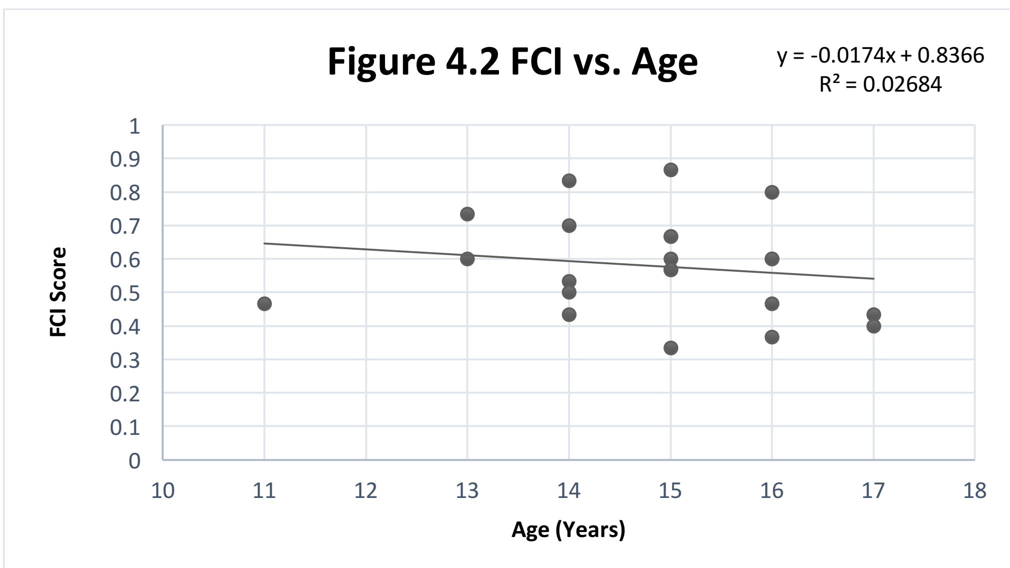
Force Concept Inventory Results

Research Question 1: Will Force Concept Inventory results be the same for students who learn from coding physics simulations as compared to students from a traditional physics course? I hypothesized that coding students would perform equally well on the FCI as compared to students from a traditional learning environment. The results from the summer coding academies provide support for this hypothesis.

Of the 32 students who participated, 22 took the FCI. The FCI was not administered to the 4 pilot students. Two additional students withdrew from the academy for personal reasons. One student left on vacation during the 2nd week and 3 others were absent on the final academy day but did not report the reason. The mean FCI score from Academy 1 was 56.4 % and the mean for Academy 2 was 59.4%. There was no statistically significant difference between the two academies. The overall average was 57.9%, which is significantly better than a 20% random guessing rate and near the 60% "conceptual threshold" described by Hestenes, Wells, and Swackhamer (1992, 161). Students who reach the conceptual threshold of 60% are more likely to succeed at more advanced topics. The minimum score was 33.3% and the maximum score was 86.7%. The median was 58.3% with a standard deviation of 14.9. Figure 4.1 shows the score distribution.



The youngest participant was 11 and the oldest was 17. There was no significant correlation between age and FCI score. Figure 4.2 shows the relationship, or lack thereof, between age and FCI score.



The sample size is too small to expect significance while analyzing subgroups, but participants were asked about their prior experience taking algebra, geometry, physics, and AP physics courses. None of the participants had taken AP physics. Results from this study indicate that past classroom experience with physics, algebra, and geometry have little or no effect on success during the coding academy. In fact, students who reported taking physics performed slightly worse on the FCI. This data is presented in Table 4.4.

Table 4.4. Courses Taken Prior to Coding Academy

Course	% Taken	Mean FCI With	Mean FCI Without	Significant at $p < 0.05$?
Physics	41	54.1	60.5	No
Algebra	82	58.5	55.0	No
Geometry	59	58.7	56.7	No

To directly address the first research question, I will provide a comparison to data from Hestenes, Wells, and Swackhamer (1992). They published data for 8 high schools and 5 colleges/universities. Standard deviations were unavailable for 4 of these institutions, so they are not included in Table 4.5. Present study data is labeled "Academy."

Table 4.5. Two-Tailed Comparison of Coding Academy and Traditional FCI Means

Location	Level	n	s	FCI Mean	t-stat	$p < 0.05$?
Academy	HS/MS	22	0.15	0.579	N/A	N/A
AZ Reg	HS	612	0.16	0.48	3.05	Yes
Wells Reg	HS	18	0.2	0.64	-1.08	No
AZ Hon	HS	118	0.19	0.56	0.52	No
Wells Hon	HS	30	0.15	0.78	-4.80	Yes
AZ AP	HS	33	0.18	0.57	0.20	No
Van Heuvelen	Univ	116	0.18	0.63	-1.43	No
AZ State Reg	Univ	139	0.18	0.63	-1.45	No
Harvard Reg	Univ	186	0.15	0.77	-5.69	Yes

In five out of eight locations where students were taught Newtonian physics in a traditional learning environment, no significant difference was found using a two-tailed test at $p < 0.05$. One high school was found to be significantly lower at $p < 0.01$. Wells

Honors and Harvard Honors were found to be significantly higher at $p < 0.01$. These results indicate that a two-week coding academy can potentially have the same FCI posttest scores as compared to a semester of traditional instruction.

After being challenged on the factors, Hestenes and Halloun (1995) emphasized that the FCI test was designed to be analyzed as a whole and not broken down into factors, especially when administered to novices. In this chapter, I have so far presented aggregated data in terms of full FCI results. However, I will delve a bit deeper into the FCI before moving to the survey data. Question 26 was answered correctly by 9% of participants and question 27 was answered correctly 95% of the time. These two questions represent the single worst and single best results from the FCI test. Notably, both questions are based on the same scenario from question 25, a woman applying a constant force to a box resulting in constant speed motion. For reference, I will reproduce questions 26 and 27 from Hestenes, Wells, and Swackhamer (1992).

26. If the woman in the previous question doubles the constant horizontal force that she exerts on the box to push it on the same horizontal floor, the box then moves:

(A) With constant speed that is double the speed "v_o" in the previous question.
(B) With constant speed that is greater than the speed "v_o" in the previous question, but not necessarily twice as great.
(C) For a while with a speed that is constant and greater than the speed "v_o" in the previous question, then with speed that increases thereafter.
(D) For a while with an increasing speed, then with constant speed thereafter.
(E) With continuously increasing speed.

27. If the woman in question 25 suddenly stops applying a horizontal force to the box, then the box will:

(A) Immediately come to a stop.
(B) Continue moving at a constant speed for a while and then slow to a stop.
(C) Immediately start slowing to a stop.
(D) Continue at a constant speed.
(E) Increase its speed for a while and then start slowing to a stop.

Since I did not interview the students, I do not specifically know what they were thinking on these two questions. However, students were apparently able to identify the presence of a net force on question 27, but not on question 26. This is something I would like to investigate further.

CLASS Results

The Colorado Learning Attitudes about Science Survey (CLASS) has eight categories including real world connections, personal interest, sense making/effort, conceptual connections, applied conceptual understanding, problem solving general, problem solving confidence, and problem solving sophistication. Survey results are measured against established expert responses on the instrument and are reported as percent favorable. In a pre/post comparison for the participants in this study, there were significant positive gains in four categories at the 95% confidence level, non-significant gains in two categories, and a non-significant decrease in two categories. These results are summarized in Table 4.6. This data includes twenty students.

Table 4.6. Two-Tailed CLASS Pre/Post Comparison

Category	Pre Academy	Post Academy	Gain/Loss	t-stat	p < 0.05?
Overall	0.6563	0.7063	0.0500	-3.85	Yes
Real World	0.8000	0.8688	0.0687	-3.58	Yes
Personal Interest	0.7708	0.8542	0.0833	-2.94	Yes
Sense Making	0.7607	0.8214	0.0607	-2.29	Yes
Conceptual Connect.	0.5875	0.6375	0.0500	-1.13	No
Applied Concept.	0.5042	0.4750	-0.0292	0.73	No
Prob. Solving Gen.	0.7536	0.7893	0.0357	-1.49	No
Prob. Solving Conf.	0.7417	0.8313	0.0896	-2.61	Yes
Prob. Solving Soph.	0.6550	0.6542	-0.0008	0.02	No

Four additional students took the pre and post surveys but were removed from the analysis because they answered question 31 incorrectly. Question 31 on the survey is used to eliminate responses of people who are not reading the statements carefully. This was the only question students asked me about while I was administering the survey. Some students joked about question 31 being so easy and some students were sincerely confused by it. Two of the students who asked me about question 31 ultimately answered it incorrectly.

In a study conducted by the developers of CLASS, Adams et al. (2006) provided data from a university physics course involving 397 participants. Despite finding significant conceptual gains with force and motion, the group's attitudes became less favorable in every category. In a follow-up study with 1380 students, they showed that although personal attitudes tend to decrease in favorability while taking a physics course, students can often identify the expert answers on the CLASS survey (Gray, Adams, Wieman, & Perkins, 2008). Academy pre/post data is reproduced in Table 4.7 along with the Adams et al. (2006) pre/post data.

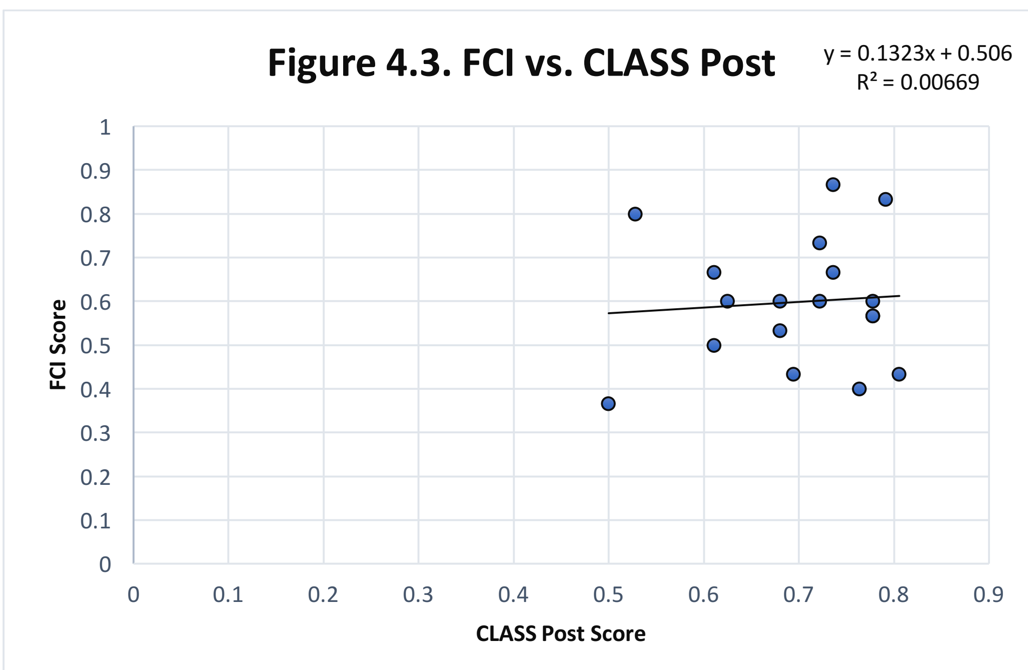
Table 4.7. Coding Academy and University Pre/Post CLASS Data

Category	Pre Academy	Pre Univ Physics	Post Academy	Post Univ Physics
Overall	0.6563	0.65	0.7063	0.59
Real World	0.8000	0.72	0.8688	0.65
Personal Interest	0.7708	0.67	0.8542	0.56
Sense Making	0.7607	0.73	0.8214	0.63
Conceptual Connect.	0.5875	0.63	0.6375	0.55
Applied Concept.	0.5042	0.53	0.4750	0.47
Prob. Solving Gen.	0.7536	0.71	0.7893	0.58
Prob. Solving Conf.	0.7417	0.73	0.8313	0.58
Prob. Solving Soph.	0.6550	0.61	0.6542	0.46

Research Question 2: How will the simulations affect student recognition of real world connections and how will personal interest in physics be affected by the coding academy? I hypothesized that collectively, student personal interest in physics and ability to recognize real world connections will increase significantly during the coding academy. The hypothesis is supported by the preceding data. The effect size was less than 10% for each category, but personal interest and real world connections did increase significantly during the academy. Given the tendency for attitudes to become less favorable after taking a physics class, these results are promising.

FCI and CLASS Relationship

No significant relationship was found between the FCI and the CLASS survey. Chart 4.3 shows the lack of correlation between FCI results and CLASS post data for individual students. Chart 4.4 shows the lack of correlation between FCI results and CLASS gains.



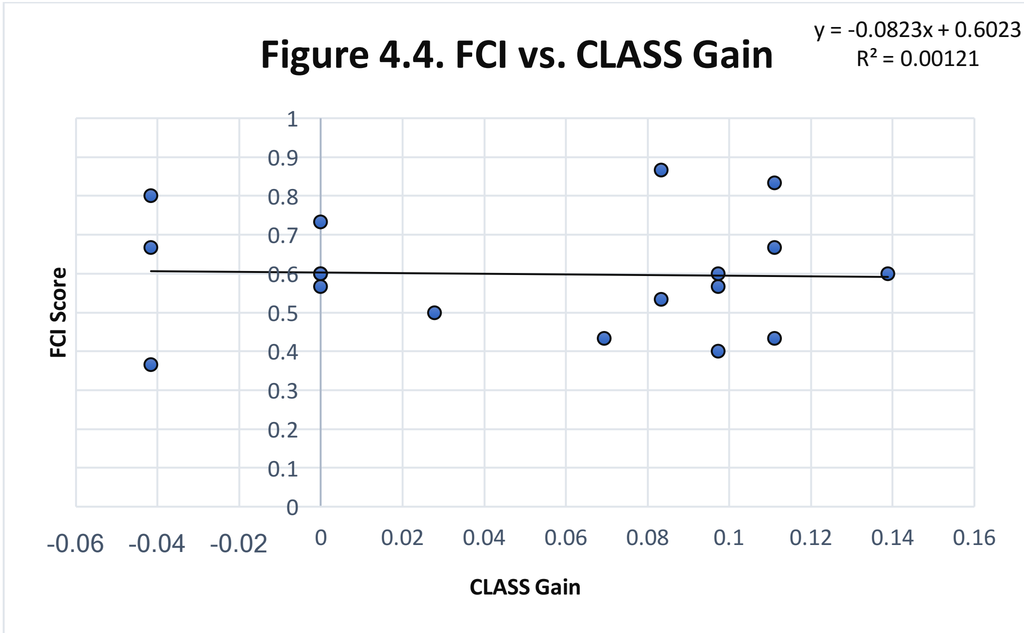


Figure 4.4. FCI vs. CLASS Gain
y = -0.0823x + 0.6023
R² = 0.00121
FCI Score
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.06
-0.04
-0.02
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
CLASS Gain

Chapter 5: Conclusion

Quick Summary

The purpose of this study was to find out if Processing sketches could be used to teach physics. To that end, I used the FCI to measure the conceptual understanding of students and the CLASS to measure their attitudinal changes regarding physics. The FCI results show that coding can be as effective as traditional instruction, but requires less instructional time. CLASS results show that student attitudes about physics become more favorable after learning to program simulations. Real world connections, personal interest, sense making/effort, and problem solving confidence all increased significantly among the participants. Overall favorability also increased significantly.

Expected Results

Consistent with Hake (1998), this study adds to the body of evidence that interactive-engagement teaching methods are effective. Although little research has been published on teaching high school physics by asking students to write simulations, this technique has many of the attributes of other interactive teaching methods. For example, launching a projectile across a classroom is not altogether different than launching a projectile across a computer screen. But with the computer program, each student gets their own experiment to customize at will. And projectiles on a computer screen can be launched in a more haphazard way without any safety risks. They can also be launched more rapidly to increase the number of observations.

Regarding attitudinal changes, I predicted that real world connections and personal interest would increase during the academies. Since the labs were completed virtually, it was possible that students would not make connections to the physical world.

But consistent with Finkelstein et al. (2005), this study showed that virtual learning can translate to real world connections. My hypothesis regarding personal interest was based on anecdotal evidence from my own experience as a classroom teacher. I had previously noticed that coding makes some students more interested in physics and does not seem to dissuade the students who were already interested. The survey data gathered during the coding academies of this study supports this view.

Unexpected Results

With a small sample size, it is not surprising when results are not statistically significant. However, I was surprised to see how small of a correlation there was between age and FCI scores. With a greater potential for prior knowledge, I expected the older students to perform better, but the small correlation was negative. In this study, younger students performed slightly better than older students. On a related note, past experience with algebra, geometry, and physics were not significantly related to FCI scores. And past experience taking a physics class was correlated with slightly lower FCI scores. This would be interesting to investigate further. I suspect the students who have taken physics may be more vulnerable to the distractors on the test, since they may recognize more of the vocabulary. Since experts score well on attitudinal and conceptual tests, I expected to see a correlation between FCI and CLASS results, but there was none. There is not enough data to be conclusive, but these results suggest that conceptual learning in physics and attitudes about the subject may be unrelated for novice students.

Coding in the Content Areas

As a practical matter, schools need a way to teach studnets how to code. For affluent school districts, it may be possible to recruit, train, and compensate staff members dedicated to teaching code for coding's sake. In districts with less financial resources, hiring new staff is a less plausible solution. What if we could teach students to code without hiring new teachers? What if those teachers were already on staff in our schools?

The results presented in Chapter 4 showed how computer programming can be an efficient method for teaching physics. But there are some bigger implications here. The students also learned about coding. If students can learn about coding and physics while gaining more favorable attitudes, perhaps they could also learn to code in chemistry or math class. Or use code to make maps in geography class. Or code interactive non-linear stories in literature classes.

Papert (1980) was ahead of his time when he suggested giving all children a computer and teaching them to code. But we are now in a position to implement Papert's vision in our schools. Reading, writing, and arithmetic are alive and well in the content areas. Now is the time to add programming to the list of fundamental skills used in the classroom.

Theoretical Implications

In chapter 1, I traced multiple paths through the literature on concepts. Experimental evidence supports various theories on concepts which often seem contradictory and are presented as competing alternatives in the literature. Yet, when mapped onto the Bayesian theory of learning, these alternatives look like different pieces of the same puzzle. I called this combination the triangulation model. The similarity-based theories, such as prototypes and exemplars, can be viewed as a cognitive ability to compare attributes of concepts stored as prior knowledge to attributes of observed evidence. When hypothetical knowledge is added to the mix, Bayes' theorem emerges, providing a mechanism for conceptual change.

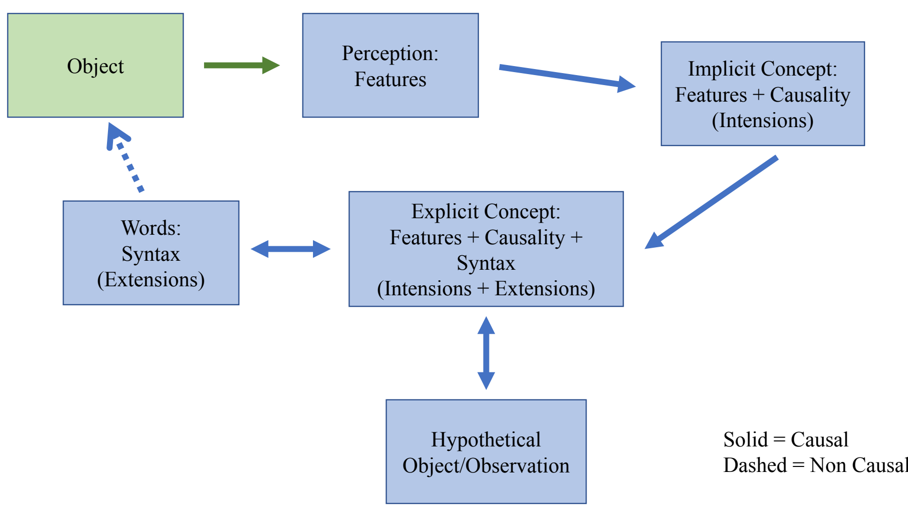
The triangulation model defines concepts as theoretical categories, which are based on exemplars, abstractions (e.g. prototypes), and causal relationships. Conceptual change occurs by splitting or merging an ontological category, changing the ontological category of a term, updating the predicates that span a category, or revising the causal relationships between or within categories (Carey, 1985; Keil, 1989; Rehder, 2003b). By combining theories that describe the properties of individual concepts, the mental organization of concepts, and the dynamics of conceptual change, the triangulation model provides a broad view of conceptual learning.

As described in Chapter 2, students arrive in physics class with a non-Newtonian view of motion (Champagne, Klopfer, & Anderson, 1980). Thus, teaching Newtonian physics requires conceptual change. For conceptual change, students must revise their ontology and often their understanding of causal relationships. To make these changes, the mind evaluates hypothetical knowledge given observational evidence and prior knowledge. Following Bayes, the key to the triangulation model is incorporating hypothetical knowledge, prior knowledge, and evidence into the same activity. When students only consider their prior knowledge, they are unlikely to accept inconsistent hypothetical knowledge in the absence of new evidence. And when students see evidence that supports a hypothesis without considering how it relates to their prior knowledge, they are at risk of forming a synthetic concept instead of a coherent conceptual understanding (Vosniadou & Skopeliti, 2013). In this experiment, coding simulations gave students everything they needed to change their mind about Newtonian physics. While developing their own physics simulations, students can readily test hypotheses against what they already know and what they are observing.

I have emphasized the distinction between simulations and physical reality. However, there is a lesser explored distinction between the simulations running in our brain and the simulations running on a computer. In a forthcoming paper, Piccinini (in press) sorts through various kinds of mental simulations on his path to naturalize intentionality. I think this type of approach has implications for learning as well. Although it would be difficult to study, it is possible there is a connection between the simulations students write in code and the simulations that run inside their brains to model the world around them. It seems likely that student's computer code could be used to evaluate their understanding of physics. It is a bigger stretch, but still plausible, that the coding method is effective because the physics simulations we run in our brain can be updated by exploring the code for computer simulations. Perhaps future research into mental simulations will elucidate this relationship.

For the processing motion study, the triangulation model provides an adequate explanation for why coding is beneficial to physics students. But in the future, I would like to take a closer look at conceptual structures. Earlier, I explored the combination of the Piccinini (2011) distinction between explicit and implicit concepts with the Fodor and Pylyshyn (2015) view on the relationship between thoughts and language. In figure 5.1, I have mapped out how concepts could have different properties depending on whether they contain associative, causal, or syntactic information. This figure also relates intensions to features and causality while extensions are assigned to syntax. The solid arrows are my proposal for which conceptual structures can be caused by others. For example, a perception can contain information about surface features which could affect a concept. But a concept cannot cause a perception. The dashed arrow indicates that a word can represent, but not cause, an object. And an explicit concept is an implicit concept with a word attached. This could be a word from a language like English or the internal syntax used by the mind. For example, we can think of an object without knowing the word for it. To fully defend this view of concepts would require a project on a grander scale than my dissertation. But since it developed during my dissertation research and I think it points towards how a more complete theory of concepts could develop, I decided to place it here at the end. After all, the end of one thing is often the beginning of another.

Figure 5.1. Conceptual Structures and Causal Connections



Appendix: Academy Projects

Project 1: Where is my mouse?

Background: The Processing window of your sketch is called a canvas. It is a grid of pixels, often many thousands of pixels. Each pixel has an (x, y) coordinate and the top left is (0, 0). The x values get larger as you go to the right. The y values get larger as you go down. By controlling the color of the pixels on the canvas, you can essentially draw anything.

Mission: To get warmed up, let's type some code to learn about the (x, y) coordinates on your canvas. Start with the sample code below and then tweak it. You can, for example, change the background color, move the text, change the text color, or resize the text. In future projects, you may find it useful to insert code like this to help you find points on the screen.

```
void setup() {
 size(800, 600);
 background(100);
}

void draw() {
 background(100);
 if (mousePressed) {
   textSize(25);
   fill(0);
   text(mouseX, 200, 200);
   text(mouseY, 270, 200);
 }
}
```

Project 2: TO TURTLE

Background: In 1980, Seymour Papert wrote a book called Mindstorms. He proposed giving all kindergarteners in 1987 their own personal computer. Papert encouraged teachers to get students to program computers rather than trying to have the computer program the students. Here's an excerpt:

> "My interest is in the process of invention of 'objects to think with,' objects in which there is an intersection of cultural presence, embedded knowledge, and the possibility for personal identification. The Turtle is a computer-controlled cybernetic animal. It exists within the cognitive minicultures of the 'LOGO environment,' LOGO being the computer language in which communication with the Turtle takes place. The Turtle serves no other purpose than of being good to program and good to think with. Some Turtles are abstract objects that live on computer screens. Others, like the floor Turtles shown in the frontispiece are physical objects that can be picked up like any mechanical toy" (p. 11).

Mission: In honor of Dr. Papert, let's design a turtle made out of code. It can be as simple or abstract as you like. But instead of LOGO, we will use Processing.

Quick Reference: The main functions in P3 look like this:

```
void setup(){
	//Code goes here.  Setup runs once.
}
void draw(){
	//Code goes here.  Draw runs repeatedly, typically 30 times per second.
}
```

2D Shape	Parameter 1	2	3	4	5	6	7	8
arc	x	y	width	height	start angle	stop angle	Mode (opt)	
ellipse	x	y	width	height				
line	x1	y1	x2	y2				
point	x	y						
quad	x1	y1	x2	y2	x3	y3	x4	y4
rect	x	y	width	height	r (opt)			
triangle	x1	y1	x2	y2	x3	y3		

You can make a custom shapes using beginShape(); followed by a list of vertices with vertex(x, y); finishing with endShape(CLOSE);.

You can color shapes with fill(r, g, b); and the outline of shapes with stroke(r, g, b);. To change the thickness of the outline, use strokeWeight(px);

See http://processing.org/reference/ for a full reference page with richer descriptions.

Project 3: Vehicles

Background: In his 1984 book *Vehicles*, Valentino Braitenberg describes a series of machines that use motors to move and sensors to interact with the world. Here is an excerpt where Braitenberg explains the design process:

> "It is also quite easy to observe the full repertoire of behavior of these machines - even if it goes beyond what we had originally planned, as it often does. But it is much more difficult to start from the outside and to try to guess internal structure just from the observation of behavior...when we analyze a mechanism, we tend to overestimate its complexity. In the uphill process of analysis, a given degree of complexity offers more resistance to the workings of our mind than it would if we encountered it downhill, in the process of invention" (p. 20).

Instead of looking at someone else's simulation, or reading their code and trying to make sense of it, we are going to use our creativity to write our own code.

Mission 1: Check out Maciek Albrecht's vehicles that were inspired by Braitenberg. Design your own vehicle and draw it with code.

Mission 2: Vehicles are often capable of moving. Follow the steps for designing a simulation and write code to sketch a simple vehicle that can move in a straight line at a constant speed. After you get the math figured out, rewrite your code using vectors.

Mission 3: Move the background function to setup() and remove it from draw(). Then we can trace the path of moving objects. As we make new projects, pay attention to how different kinds of motion make different kinds of paths.

Quick Reference: When you are working with 2 or more dimensions, vectors are incredibly helpful. See https://processing.org/reference/PVector.html for the full story. But in P3, you declare and initialize a vector like this:

```
PVector position = new PVector(5, 8);
```

This will reduce the total number of lines, because vectors can store the x and y coordinates together. When you need to refer to the x-component only, simply type .x after the name of the vector:

```
position.x = 2;
```

If your object goes off the screen and you want it to show up on the other side, try something like this:

```
if( position.x > width ){
position.x = 0;
}
```

Math Operations	Addition	Subtraction	Multiplication	Division
Arithmetic	x += 2;	x -= 2;	x *= 2;	x /= 2;
Vector	position.add(x, y);	position.sub(x, y);	position.mult(2);	position.div(2);

Project 4: Acceleration

Background: In 1678, Newton gave science a detailed theory of motion that is strongly supported by evidence. In the absence of an unbalanced force, objects have a constant velocity (no speeding up, slowing down, or changing direction). Unbalanced forces cause an object to speed up, slow down, or change direction. Changes in velocity are always in the same direction as an unbalanced force. The amount of change in velocity is directly proportional to the size of the force and inversely proportional to the mass of the object. Lastly, every force in the universe has a companion force of equal strength, but in the opposite direction. Keep Newton in mind as you make simulations of motion. If you can represent Newtonian ideas with code, your simulations will look more like real motion.

Mission 1: Last time, we made a position vector to control the location of our objects. Now, let's make a velocity vector to control the speed and direction. If you can figure out how to code acceleration, your objects will be able to speed up and slow down. Do not forget to turn off the background sometimes so you can see the motion paths. It really helps you visualize how stuff moves.

Mission 2: Make a simulation of an object falling in a gravitational field. Can you simulate the gravity on multiple planets in one sketch?

Mission 3: Make a simulation of an object falling through a fluid like air or water. If your code works, now the objects will have a terminal velocity. Can you simulate two different materials in one sketch? For example, imagine a ball falling through the air and then slowing down as it lands in some water. Air friction is surprisingly complicated to calculate. But for the purpose of a simulation, air drag at low speeds is typically proportional to the velocity (and in the opposite direction). At high speeds, air drag is typically proportional to the velocity squared (still in the opposite direction). So make sure your model of air friction is responsive to the velocity vector.

Quick Reference: During a simulation, it is cool to display the vector values on the screen while objects are moving around. If you put something in quotes in the text function, P3 will display exactly what you type. If you enter a variable without quotes, P3 will display the current value of the variable stored in memory. Here are some sample lines for the horizontal component of the velocity vector:

```
text("Horizontal Velocity", 100, 80);
text(velocity.x, 100, 100);
```

Project 5: Projecting Projectiles

Background: We now know the position of an object can be changed by adding the velocity vector. And the velocity can be changed by adding the acceleration vector. Galileo figured out that projectiles are a combination of constant velocity in one dimension with accelerated motion in the other dimension. He also figured out a super sweet pattern with accelerated objects (see below). This table uses the acceleration due to gravity on earth, but the same pattern emerges with any constant acceleration.

Galileo Pattern

Total Time (s)	1	2	3	4	5	6	7
Total Dist (m)	9.8	29.4	49	68.6	88.2	107.8	127.4
Dist / 9.8	1	3	5	7	9	11	13
Cumulative	1	4	9	16	25	36	49

Mission 1: Make a simulation with position, velocity, and acceleration vectors. Experiment with combinations of constant velocity in one dimension and constant acceleration in the other. Use positive and negative values of each to see what happens. Try to connect the motions you see on the screen to motions you have observed in real life.

Mission 2: Turn off the background in draw() and experiment with making cool curved paths. Or use transparency instead of background to see motion blue…

```
fill(0, 50);
rect(0, 0, width, height);
```

Mission 3: Launch projectiles on a curved path from one corner and hit other corners of your canvas.

Project 6: Collisional Colliding

Background: There are two types of collisions, elastic and inelastic. With perfectly elastic collisions, the kinetic energy is constant. With inelastic collisions, some of the kinetic energy is converted to other forms of energy, so the speed of an object tends to be reduced after a collision.

Mission 1: Make an object bounce off the wall (edge of canvas) without losing any speed (elastic). Then Make an object bounce off the wall and subsequently lose speed (inelastic).

Mission 2: Simulate two objects bouncing off each other.

Quick Reference: If you want an object to bounce off the edge, try something like this:

```
if( position.x > width ){
velocity.x *= -1;
}
```

If you want fast acceleration without fast velocity, make a speed limit like…velocity.limit(15);
And if it comes up, try noCursor(); to hide your mouse cursor.

Project 7.1: Rocketeer Part 1

Background: Rockets in outer space can move around without air drag (since there is very little air). They can move in any direction, but like all objects, can only accelerate in one direction at a time (based on the net force). Rockets moving through air do experience air drag. The faster they go, the greater the air impedes their motion. Air friction is always in the opposite direction of the velocity and therefore tends to reduce the speed of moving objects.

Mission 1: Sketch a rocket whose acceleration can be controlled with the keyboard.

Mission 2: Add a ground to the sketch for the rocket to land on.

Mission 3: Add some more physics. Possible ideas...air friction, ground friction, gravity.

Mission 4: Introduce some background changes as the rocket increases elevation. For example, maybe the sky could go from being blue to black as you enter space. Or maybe the ground could disappear as the rockets get farther from it.

Quick Reference: Based on our interactivity project, this type of control system seems to work well...

```
if (keyPressed) {
   if (keyCode == UP) {
     velocity.y -= acceleration.y;
   }
   if (keyCode == DOWN) {
     velocity.y += acceleration.y;
   }
   if (keyCode == LEFT) {
     velocity.x -= acceleration.x;
   }
   if (keyCode == RIGHT) {
     velocity.x += acceleration.x;
   }
 }
```

Project 7.2: Rocketeer Part 2

Mission 1: Customize the backgrounds for each stage, including adding new shapes. See comments in code for where to add them. Also, include acceleration controls. See previous project if you forgot how.

Mission 2: Add interactive elements to stages, customize mover, and change acceleration for each stage.

```
int stage = 0;
PVector position = new
PVector(300, 500);
PVector velocity = new PVector (0,
0);
PVector acceleration = new
PVector(0.1, 0.1);

void setup() {
  size(1000, 700);
  noStroke();
}

void draw() {
  if (stage == 0) {
    background(200);
    fill(0);
    textSize(25);
    text("Make Gamer", 200, 200);
    textSize(15);
     text("Press Enter to Enter", 200,
300);
     text("Then Use Arrow Keys",
200, 350);
    fill(200);
    velocity.set(0, 0);
    if (keyPressed && key ==
ENTER) {
      stage = 1;
    }
  } else if (stage == 1) {
    background(0, 0,
position.y*255/(9*height/10));
//fading bground
     //add background shapes here
    fill(50, 0, 50);           //fill for
ground
    if (position.x < 0) {
      position.x -= velocity.x;
      velocity.x *= -.5;
//inelastic bounce
    }
  } else if (stage == 2) {
    //add new background
    //add new background shapes
    //add new fill for ground
  } else if (stage == 3) {
    background(200);
    //add new background shapes
    //add new fill for ground
    if (position.x > width) {
      position.x -= velocity.x;
      velocity.x *= -.5;         //inelastic bounce
    }
  }
  pushMatrix();
  if (position.y < height/6) {
    translate(0, height/10-
position.y+height/15);   //ground moves
down when you go up
  }
  rect(0, 9*height/10, width, height/10);
//Ground
  popMatrix();

  if (position.x > width) {
    position.x = 0;
    stage += 1;            //increase stage on
right
  }
  if (position.x < 0) {
    position.x = width;
    stage -=1;                  //decrease stage on
left
  }
  if (position.y > 9*height/10) {
    position.y = 9*height/10;
    velocity.mult(.9);
  }
  if (position.y < height/15) {
    position.y = height/15;
    velocity.y*=0.1;
  }

//Enter your acceleration controls here

  position.add(velocity);     //velocity
Equation
  fill(200, 0, 0);
  ellipse(position.x, position.y, 25, 25);
//Mover
}
```

Project 8: Bouncer

Background: In the collision project, we learned how to bounce objects based on their position. Now, let's learn how to make them bounce off other objects based on color.

Mission 1: Add bouncer objects to your previous projects, especially the rocket sketch.

Mission 2: Make these objects do something other than bouncing. For example, maybe if you hit one, game over.

Quick Reference: First, at the very top, declare a color variable to work with…

```
color bounce = color(0, 0, 0);
```

Then, in one of your stages, draw a shape with a unique red fill parameter…

```
fill(1, 200, 0);
rect(400, 400, 50, 50);
```

Lastly, get the pixel color at the location of your object. In this example, the get() function requires us to round position variables, since they are float variables and we need int variables. Be sure to put this code in before you draw an object at the location. Otherwise, it will just get the color of your object and not of the background shapes. If you move inside a shape with the unique red parameter, in this case any shape with red set to 1 or 251, the object will bounce off with a significant force. You can do the same thing with the green() or blue() functions.

```
bounce = get(round(position.x), round(position.y));
  if (red(bounce) == 1 || red(bounce) == 251) {
    velocity.mult(-3);
  }
```

Project 9: Cannon

Background: This is a follow up to our projectile project. This time, we are going to add interactivity to control the initial velocity of the launch. After projectiles are launched, the only forces are gravity and air drag (if air is present). The reference code is a simulated cannon that sets the launch angle based on mouseY and the initial speed based on mouseX. It launches when you click the mouse. The projectile will launch at the angle shown by the line in the lower left corner. To reset, click the mouse again.

Mission 1: Tweak the visuals of the reference code. And try turning off the background in draw(). You can make some nice parabolas with this one.

Mission 2: Tweak the numbers in the reference code. Avoid changing math related radians and degrees. You can change the speed of a projectile, but you cannot change the number of radians in a degree. By the way, Processing (and most mathematicians) use radians instead of degrees. The conversion is pretty straightforward. 180 degrees is equal to 3.14 radians One more thing. You will notice a % symbol in the code. It does not mean percent. It stands for modulo, which divides the first number by the second number and then returns the remainder. For example, 10 % 2 = 0, 10 % 3 = 1, and 10 % 4 = 2.

Mission 3: Add air friction to the simulation.

Mission 4: Add a target to the simulation.

Quick Reference:

```
PVector cannon;
float radians;
float degrees = 0;
int integer = 0;
PVector position = new PVector(0, 0);
PVector velocity = new PVector(0, 0);
PVector acceleration = new PVector(0, .098);

void setup() {
 fullScreen();
 background(0);
}
void draw() {
 background(0);
 if (integer % 2 < 1) {
  fill(0);
  noStroke();
  rect(100, 60, 200, 50);
  fill(255);
  position.set(0, height);
  radians = 3.14*degrees/180;
  cannon = PVector.fromAngle(radians);
  degrees = 90*mouseY/height+270;
  cannon.setMag(mouseX/2);
  text("degrees", 100, 80);
  text(360-degrees, 100, 100);
  text("x", 180, 80);
  text(cannon.x, 180, 100);
  text("y", 240, 80);
  text(abs(cannon.y), 240, 100);
  stroke(0, 200, 0);
  strokeWeight(10);
  pushMatrix();
  translate(0, height);
  line(0, 0, cannon.x, cannon.y);
  popMatrix();
  velocity.x = cannon.x/30;
  velocity.y = cannon.y/30;
 }
 if (integer % 2 > 0) {
  noStroke();
  fill(random(150, 255), 0, 0);
  ellipse(position.x, position.y, 20, 20);
  position.add(velocity);
  velocity.add(acceleration);
 }
}
void mouseClicked() {
 integer++;
}
```

Project 10: Empire State Coin Drop

Background: There is a myth that a penny dropped from the Empire State Building would be fatal if it hit a passerby on the sidewalk below. This turns out to be false, but it is an interesting idea for an experiment. Since we do not have an Empire State Building and it is not a particularly safe experiment anyway, let's simulate this event.

Mission 1: Sketch the Empire State Building and add a projectile motion simulator to the top.

Mission 2: Add a building across the street so your projectile will bounce off.

Mission 3: Add a target on the ground.

Quick Reference: The Empire State Building has 6,500 windows. You do not need to draw all of them, or even a quarter of them, but drawing windows one by one is a tedious process. Luckily, we can use a for loop to solve the problem. A single for loop can draw a line of windows like this...

```
for (int x = 0; x < 7; x++) {
  rect(100+x*50, 200, 20, 30);
}
```

Use a nested for loop to make a full grid...

```
for (int x = 0; x < 7; x++) {
  for (int y = 0; y < 5; y++) {
    rect(100+x*50, 200+100*y, 20, 30);
  }
}
```

Project 11: Advanced

Background: There are lots of ways to proceed from here. Below, you will see three missions to help you continue your coding journey.

Mission 1: There are more 3D Primitives in P5.js, which you should check out too. But P3 has two important volumetric shapes, sphere and box. First, be sure to add "P3D" to the size function like this…

```
size(800, 600, P3D);
```

Next, box takes three parameters for the (x, y, z) dimensions. And sphere takes one parameter for radius. Use the translate and rotate functions to move 3D shapes.

Mission 2: Sign up for an account on Open Processing and save any code you want to keep. Switch to Processing.js mode to use P3 syntax.

Mission 3: In programming, classes store a set of functions with relevant variables and parameters. For example, if you design a car object that will drive under certain conditions, you may want to make a car class. This will give you the ability to quickly make copies of the car without rewriting all the functions. The basics are available here…

https://processing.org/reference/class.html

https://processing.org/reference/Object.html

Additional Resources:

www.openprocessing.org

http://shiffman.net/

http://natureofcode.com/book/

https://www.youtube.com/user/shiffman

http://formandcode.com/

https://www.kadenze.com/courses/introduction-to-programming-for-the-visual-arts-with-p5-js/info

https://www.kadenze.com/courses/the-nature-of-code/info

References

Abelson, H. & diSessa, A. (1980). *Turtle Geometry: The Computer as a Medium for Exploring Mathematics*. Cambridge: The MIT Press.

Adams, W. K., Perkins, K. K., Podolefsky, N. S, Dubson, M., Finkelstein, N. D., and Wieman, C. E (2006). New Instrument for Measuring Student Beliefs about Physics and Learning Physics: The Colorado Learning Attitudes about Science Survey. *Physical Review Special Topics - Physics Education Research 2*, 010101. doi: https://doi.org/10.1103/PhysRevSTPER.2.010101

Attneave, F. (1957). Transfer of Experience with a Class-Schema to Identification-Learning of Patterns and Shapes. *Journal of Experimental Psychology*, 54(2), 81-88.

Braitenberg, V. (1984). *Vehicles, experiments in synthetic psychology*. Cambridge: The MIT Press.

Brewe, E., Kramer, L., & O'Brien, G. (2009). Modeling instruction: Positive attitudinal shifts in introductory physics measured with CLASS. *Physics Review Special Topics - Physics Education Research*, 5(1), 013102. doi:10.1103/PhysRevSTPER.5.013102

Buchsbaum, D., Bridgers, S., Skolnick Weisberg, D., & Gopnik, A. (2012). The power of possibility: causal learning, counterfactual reasoning, and pretend play. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 367(1599), 2202–2212. doi:10.1098/rstb.2012.0122

Bullock, Gelman, & Baillargeon (1982). The Development of Causal Reasoning. In W. Friedman (Ed.), *The Developmental Psychology of Time*. New York: Academic Press.

Buffler, A., Pillay, S., Lubben, F., & Fearick, R. (2008). A model-based view of physics for computation activities in the introductory physics course. *American Journal of Physics*, 76(4), 431-437. doi:10.1119/1.2835045

Caballero, M.D., Kohlmyer, M.A., & Schatz, M.F. (2012). Implementing and assessing computations modeling in introductory mechanics. *Physical Review Special Topics - Physics Education Research*, 8(1), 020106. doi:10.1103/PhysRevSTPER.8.020106

Cain, S. & Laird, M. (2011). *The Fundamental 5.* Self published via CreateSpace.

Carey, Susan. (1985). *Conceptual change in childhood*. Cambridge, Massachusetts, USA: The MIT Press.

Champagne, A.B., Klopfer, L.E., & Anderson, J.H. (1980). Factors influencing the learning of classical mechanics. *American Journal of Physics*, 48(12), 1074-1079. doi:10.1119/1.12290

Chi, M. T. H. (2013). Two Kinds and Four Sub-Types of Misconcieved Knowledge, Ways to Change it, and the Learning Outcomes. In S. Vosniadou (Ed.), *International Handbook of Research on Conceptual Change*. New York: Routledge.

Chi, M.T.H., Feltovich, P.J., & Glaser, R. (1981). Categorization and Representation of Physics Problems by Experts and Novices. *Cognitive Science*, 5(2), 121-152. doi:10.1207/s15516709cog0502_2

Chi, M. T. H., & Slotta, J. D. (1993). The Ontological Coherence of Intuitive Physics. *Cognition and Instruction*, 10(2/3), 249-260. Retrieved from: http://www.jstor.org/stable/3233728

Coletta, V.P., & Phillips, J.A., (2005). Interpreting FCI Scores: Normalized gain, preinstruction scores, and scientific reasoning ability. *American Journal of Physics*, 73(12), 1172-1182. doi:10.1119/1.2117109

diSessa, A. A., (1993). Toward an Epistemology of Physics. *Cognition and Instruction*, 10(2/3), 105-225.

diSessa, A. A. (2013). A Bird's-Eye View of the "Pieces" vs. "Coherence" Controversy (from the "Pieces" Side of the Fence). In S. Vosniadou (Ed.), *International Handbook of Research on Conceptual Change*. New York: Routledge.

Elby, A. (2001). Helping physics students learn how to learn. *American Journal of Physics*, 69(S1), S54-S64. doi:10.1119/1.1377283

Finkelstein, N.D., Adams, W.K., Keller, C.J., Kohl, P.B., Perkins, K.K., Podolefsky, N.S., & Reid, S. (2005). When learning about the real world is better done virtually: A study of substituting computer simulations for laboratory equipment. *Physical Review Special Topics - Physics Education Research*, 1, 010103. doi:10.1103/PhysRevSTPER.1.010103

Fodor, J. (1972). Some reflections on L.S. Vygotsky's Thought and Language. *Cognition*, 1(1), 85-95.

Fodor, J. A. (2008). *LOT 2: The language of thought revisited*. Oxford, England: Clarendon Press.

Fodor, J., & Pylyshyn, Z. (2015). *Minds without Meanings: An Essay on the Content of Concepts*. Cambridge, Massachusetts, USA: MIT Press.

Gelman S.A., & Markman, E.M. (1987). Young Children's Inductions from Natural Kinds: The Role of Categories and Appearances. *Child Development*, 58(6), 1523-1541. Retrieved from: http://www.jstor.org/stable/1130693

Gopnik, A., Glymour, C. Sobel, D. M., Schulz, L. E., & Kushnir, T. (2004). A Theory of Causal Learning in Children: Causal Maps and Bayes Nets. *Psychological Review*, 111(1) 3-32.

Gopnik, A., & Meltzoff, A. N. (1997). *Words, thoughts, and theories*. Cambridge, Mass: MIT Press.

Gopnik, A., & Tenenbaum, J. B. (2007). Bayesian networks, Bayesian learning and cognitive development. *Developmental Science*, 10(3), 281-287. doi:10.1111/j.1467-7687.2007.00584.x

Gopnik, A., & Wellman, H. M. (2012). Reconstructing Constructivism: Causal Models, Bayesian Learning Mechanisms, and the Theory Theory. *Psychological Bulletin*, 138(6), 1085-1108. doi:10.1037/a0028044

Gray, K.E., Adams, W.K., Wieman, C.E., & Perkins, K.K. (2008). Students know what physicists believe, but they do not agree: A study using the CLASS survey. *Physical Review Special Topics - Physics Education Research*, 4(1), 020106. doi:10.1103/PhysRevSTPER.4.020106

Griffiths, T.L., & Tenenbaum, J.B. (2009). Theory-Based Causal Induction. *Psychological Review*, 116(4). 661-716. doi: 10.1037/a0017201

Halloun, I.A., & Hestenes, D. (1985a). The initial knowledge state of college physics students. *American Journal of Physics*, 53(11), 1043-1055. doi:10.1110/1.14030

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

Halloun, I., & Hestenes, D. (1998). Interpreting VASS Dimensions and Profiles for Physics Students. *Science & Education*, 7(6), 553-577. doi: 10.1023/A:1008645410992

Hake, R.R. (1998). Interactive-engagement versus traditional methods: A six-thousand-student survey of mechanics test data for introductory physics courses. *American Journal of Physics*, 66(1), 64-74. doi: 10.1119/1.18809

Hammer, D. (1994). Epistemological Beliefs in Introductory Physics. *Cognition and Instruction*, 12(2), 151-183. Retrieved from: http://www.jstor.org/stable/3233679

Hampton, J.A. (1982). A demonstration of intransitivity in natural categories. *Cognition*, 12(2), 151-164.

Heller, P., & Huffman, D. (1995). Interpreting the force concept inventory: A reply to Hestenes and Halloun. *The Physics Teacher*, 33(8), 503-503. doi: 10.1119/1.2344279

Henderson, C. (2002). Common Concerns About the Force Concept Inventory. *The Physics Teacher*, 40(9). 542-547. doi: 10.1119/1.1534822

Hestenes, D. (1996). Modeling methodology for physics teachers. Paper presented at The changing role of physics departments in modern universities: International Conference on Undergraduate Physics, College Park, MD. Retrieved March 12, 2017, from: http://modeling.asu.edu/R&E/ModelingMeth-jul98.pdf

Hestenes, D., & Halloun, I. (1995). Interpreting the force concept inventory: A response to March 1995 critique by Huffman and Heller. *The Physics Teacher*. 33(8). 502-502. doi: 10.1119/1.2344278

Hestenes, D., & Wells, M. (1992). A mechanics baseline test. *The Physics Teacher*, 30(3). 159-166. doi: 10.1119/1.2343498

Hestenes, D., Wells, M., & Swackhamer, G. (1992). Force concept inventory. *The Physics Teacher*, 30(3), 141-158. doi:10.1119/1.2343497

Hudson, H.T., & McIntire, W.R. (1977). Correlation between mathematical skills and success in physics. *American Journal of Physics*, 45(5), 470-471. doi:10.1119/1.10823

Huffman, D., & Heller, P. (1995). What does the force concept inventory actually measure?. *The Physics Teacher*, 33(3), 138-143. doi: 10.1119/1.2344171

Keil, F.C. (1981). Constraints on Knowledge and Cognitive Development. *Psychological Review*, 88(3), 197-227.

Keil, F.C. (1989) *Concepts, Kinds and Cognitive Development.* Cambridge, Massachusetts, USA: The MIT Press.

Kolodiy, G.O. (1988). Projectile motion with the turtle. *The Physics Teacher*, 26(6), 374-377. doi:10.1119/1.2342549

Lindsey, B.A., Hsu, L., Sadaghiani, H., Taylor, J.W., & Cummings, K. (2012). Possitive attitudinal shifts with the Physics by Inquiry curriculum across multiple implementations. *Physical Review Special Topics - Physics Education Research*, 8(1), 010102. doi: 10.1103/PhysRevSTPER.8.010102

Marušić, M., & Sliško, J. (2012). Effects of two different types of physics learning on the results of CLASS test. *Physical Review Special Topics - Physics Education Research*, 8(1) 010107. doi: 10.1103/PhysRevSTPER.8.010107

McCloskey, M., Caramazza, A., & Green, B. (1980). Curvilinear Motion in the Absence of External Forces: Naive Beliefs about the Motion of Objects. *Science*, 210(4474), 1139-1141.

McCloskey, M., & Glucksberg, S. (1978). Natural categories: Well defined or fuzzy sets?. *Memory & Cognition*, 6(4), 462-472.

McCloskey, M., & Kohl, D. (1983). Naive Physics: The Curvilinear Impetus Principle and Its Role in Interactions With Moving Objects. *Journal of Experimental Psychology*, 9(1), 146-156.

McCloskey, M., Washburn, A., & Felch, L. (1983). Intuitive Physics: The Straight-Down Belief and Its Origin. *Journal of Experimental Psychology*, 9(4), 636-649.

Medin, D. L., & Schaffer, M. M. (1978). Context Theory of Classification Learning. *Psychological Review*, 85(3), 207-238.

Milner-Bolotin, M., Antimirova, T., Noack, A., & Petrov, A. (2011). Attitudes about science and conceptual physics learning in university introductory physics courses. *Physics Review Special Topics - Physics Education Research*, 7(1), 020107, doi: 10.1103/PhysRevSTPER.7.020107

Missouri Department of Elementary and Secondary Education. Missouri Comprehensive Data. Retrieved August 29, 2017, from https://mcds.dese.mo.gov/quickfacts/SitePages/DistrictInfo.aspx

Murphy, G. L. (2002). *The big book of concepts*. Cambridge, Mass: MIT Press.

Murphy, G. L., & Medin, D. L. (1985). The Role of Theories in Conceptual Coherence. *Psychological Review*, 92(3), 289-316.

Nosofsky, R. M. (1986). Attention, Similarity, and the Identification-Categorization Relationship. *Journal of Experimental Psychology*, 115(1) 39-57.

Nosofsky, R. M. (2000). Exemplar Representation Without Generalization? Comment on Smith and Minda's (2000) "Thirty Categorization Results in Search of a Model". *Journal of Experimental Psychology*, 26(6), 1735-1743.

Papert, S. A. (1980/1993). *Mindstorms: Children, Computers, and Powerful Ideas.* New York, New York, USA: Basic Books, Inc.

PhET Interactive Simulations. (2017). Retrieved February 28, 2017, from https://phet.colorado.edu/

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference (Rev. Ed.)*. San Francisco, California, USA: Morgan Kaufmann Publishers Inc.

Pearl, J. D. (2000). *Causality: Models, Reasoning, and Inference*. New York, New York, USA: Cambridge University Press.

Piccinini, G., & Scott, S. (2006). Splitting Concepts. *Philosophy of Science*, 73(4), 390-409. Retrieved from: http://www.jstor.org/stable/10.1086/516806

Piccinini, G. (2011). Two Kinds of Concept: Implicit and Explicit. *Dialogue*, 50(1), 179-193. doi:10.1017/S0012217311000187

Posner, M. I., & Keele, S. W. (1968). On the Genesis of Abstract Ideas. *Journal of Experimental Psychology*, 77(3), 353-363.

Piccinini, G. (in press). Nonnatural Mental Representation. In J. Smortchkova, K. Dolega, T. Schlicht (Eds.), *What are Mental Representations?* Oxford: Oxford University Press.

Posner, G. J., Strike, K. A., Hewson, P.W., & Gertzog, W.A. (1982). Accommodation of a Scientific Conception: Toward a Theory of Conceptual Change. *Science Education*, 66(2), 211-227.

Pyatt, K., & Sims, R. (2012). Virtual and Physical Experimentation in Inquiry-Based Science Labs: Attitudes, Performance and Access. *Journal of Science Education and Technology*, 21(1), 133-147. doi:10.1007/s10956-011-9291-6

Reas, C., & Fry, B. (2006). Processing: programming for the media arts. *AI & Society*, 20(4), 526-538. doi:10.1007/s00146-006-0050-9

Reas, C., & Fry, B. (2014). *Processing: A Programming Handbook for Visual Designers and Artists* (2nd Ed.). Cambridge: The MIT Press.

Redish, E.F., Saul, J.M., & Steinberg, R.N. (1998). Student expectations in introductory physics. *American Journal of Physics*, 66(3), 212-224. doi:10.1119/1.18847

Reed, S. K. (1972). Pattern Recognition and Categorization. *Cognitive Psychology*, 3(3), 382-407.

Rehder, B. (2003a). A Causal-Model Theory of Conceptual Representation and Categorization. *Journal of Experimental Psychology*, 29(6), 1141-1159. doi:10.1037/0278-7393.29.6.1141

Rehder, B. (2003b). Categorization as causal reasoning. *Cognitive Science*, 27(5), 709-748. doi:10.1016/S0364-0213(03)00068-5

Rehder, B. (2006). When similarity and causality compete in category-based property generalization. *Memory and Cognition*, 34(1), 3-16.

Rehder, B., & Hastie, R. (2001). Causal Knowledge and Categories: The Effects of Causal Beliefs on Categorization, Induction, and Similarity. *Journal of Experimental Psychology: General*, 130(3), 323-360. doi:10.1037//0096-3445.130.3.323

Reiner, M., Slotta, J. D., Chi, M. T. H., & Resnick, L. B. (2000). Naive Physics Reasoning: A Commitment to Substance-Based Conceptions. *Cognition and Instruction*, 18(1), 1-34.

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

Rosch, E. (1975). Cognitive Representations of Semantic Categories. *Journal of Experimental Psychology*, 104(3). 192-233.
Savinainen, A., & Scott, P. (2002). Using the Force Concept Inventory to monitor student learning and to plan teaching. *Physics Education*, 37(1), 53-58. Retrieved from: http://iopscience.iop.org/0031-9120/37/1/307

Savinainen, A., & Viiri, J. (2007). The Force Concept Inventory as a Measure of Students' Conceptual Coherence. *International Journal of Science and Mathematics Education*, 6(4), 719-740. doi:10.1007/s10763-007-9103-x

Scalise, K., Timms, M., Moorjani, A., Clar, L., Holtermann, K., & Irvin, P.S. (2011). Student Learning in Science Simulations: Design Features That Promote Learning Gains. *Journal of Research in Science Teaching*, 48(9), 1050-1078. doi:10.1002/tea.20437

Schulz, L.E., & Bonawitz, E.B. (2007). Serious Fun: Preschoolers Engage in More Exploratory Play When Evidence Is Confounded. *Developmental Psychology*, 43(4), 1035-1050. doi: 10.1037/0012-1649.43.4.1045

Semsar, K., Knight, J.K., Birol, G., & Smith, M.K. (2011). The Colorado Learning Attitudes about Science Survey (CLASS) for Use in Biology. *Life Science Education*, 10(3), 268-278. doi: 10.1187/cbe.10-10-0133

Shiffman, D.S. (2012). *The Nature of Code: Simulating Natural Systems with Processing*. New York: Author.

Slotta, J.D., Chi, M.T.H., & Joram, E. (1995). Assessing Students' Misclassifications of Physics Concepts: An Ontological Basis for Conceptual Change. *Cognition and Instruction*, 13(3), 373-400.

Smith, J. D. (2002). Exemplar Theory's Predicted Typicality Gradient Can Be Tested and Disconfirmed. *Psychological Science*, 13(5), 437-442.

Smith, J. D., & Minda, J.P. (1998). Prototypes in the Mist: The Early Epochs of Category Learning. *Journal of Experimental Psychology*, 24(6), 1411-1436.

Smith, J. D., & Minda, J.P. (2000). Thirty Categorization Results in Search of a Model. *Journal of Experimental Psychology*, 26(1), 30-27.

Steinberg, R.N. (2000). Computers in teaching science: To simulate or not to simulate?. *American Journal of Physics*, 68(S1), S37-S41. doi:10.1119/1.19517

Stewart, J., Griffin, H., & Stewart, G. (2007). Context sensitivity in the force concept inventory. *Physics Review Special Topics - Physics Education Research*, 3(1), 010102. doi:10.1103/PhysRevSTPER.3.010102

Tao, P., & Gunstone, R.F. (1999). The Process of Conceptual Change in Force and Motion during Computer-Supported Physics Instruction. *Journal of Research in Science Teaching*, 36(7), 859-882.

Vosniadou, S. (2013). Conceptual Change in Learning and instruction: The Framework Theory Approach. In S. Vosniadou (Ed.), *International Handbook of Research on Conceptual Change*. New York: Routledge.

Vosniadou, S., & Skopeliti, I. (2014). Conceptual Change from the Framework Theory Side of the Fence. *Science and Education*, 23(7), 1427-1445. doi:10.1007/s11191-013-9640-3

Vosniadou, S., Skopeliti, I., & Ikospentaki, K. (2004). Modes of knowing and ways of reasoning in elementary astronomy. *Cognitive Development*, 19(2), 203-222. doi: 10.1016/j.cogdev.2003.12.002

Waldmann, M.R., & Holyoak, K.J. (1992). Predictive and Diagnostic Learning Within Causal Models: Asymmetries in Cue Competition. *Journal of Experimental Psychology: General*, 121(2), 222-236.

Wieman, C.E., Adams, W.K., & Perkins, K.K. (2008). PhET: Simulations that Enhance Learning. *Science*, 322(5902), 682-683.

Wieman, C.E., Adams, W.K., Loeblein, P., & Perkins, K.K. (2010). Teaching Physics Using PhET Simulations. *The Physics Teacher*, 48(4), 225-227. doi:10.1119/1.3361987

Wilcox, B.R., & Lewandowski, H.J. (2016). Students' epistemologies about experimental physics: Validating the Colorado Learning Attitudes about Science Survey for experimental physics. *Physical Review Special Topics - Physics Education Research*, 12(1), 010123. doi: 10.1103/PhysRevPhysEducRes.12.010123

Windschitl, M., & Andre, T. (1998). Using Computer Simulation to Enhance Conceptual Change: The Roles of Constructivist Instruction and Student Epistemological Beliefs. *Journal of Research in Science Teaching*, 35(2) 145-160.

Zacharia, Z. (2003). Beliefs, Attitudes, and Intentions of Science Teachers Regarding the Educational Use of Computer Simulations and Inquiry-Based Experiments in Physics. *Journal of Research in Science Teaching*, 40(8), 792-823. doi:10.1002/tea.10112

Zacharia, Z.C. (2007). Comparing and combining real and virtual experimentation: an effort to enhance students' conceptual understanding of electrical circuits. *Journal of Computer Assisted Learning*, 23(2), 120-132. doi:10.1111/j.1365-2729.2006.00215.x

Zacharia, Z.C., Olympiou, G., & Papaevripidou, M. (2008). Effects of Experimenting with Physical and Virtual Manipulatives on Students' Conceptual Understanding in Heat and Temperature. *Journal of Research in Science Teaching*, 45(9), 1021-1035. doi:10.1002/tea.20260

Zwickl, B.M., Hirokawa, T., Finkelstein, N., & Lewandowski, H.J. (2014). Epistemology and expectations survey about experimental physics: Development and initial results. *Physics Review Special Topics - Physics Education Research*, 10(1), 010120. Doi: 10.1103/PhysRevSTPER.10.010120