education sciences

MDPI

Article

Ants Go Marching—Integrating Computer Science into Teacher Professional Development with NetLogo

Mike Borowczak [1,*] and Andrea C. Burrows [2]

1 Department of Computer Science, College of Engineering and Applied Science, University of Wyoming, 1000 E University Ave, Laramie, WY 82071, USA
2 School of Teacher Education, College of Education, University of Wyoming, 1000 E University Ave, Laramie, WY 82071, USA; Andrea.Burrows@uwyo.edu
* Correspondence: Mike.Borowczak@uwyo.edu

Received: 1 February 2019; Accepted: 20 March 2019; Published: 26 March 2019

check for updates

Abstract: There is a clear call for pre-collegiate students in the United States to become literate in computer science (CS) concepts and practices through integrated, authentic experiences and instruction. Yet, a majority of in-service and pre-service pre-collegiate teachers (instructing children aged five to 18) lack the fundamental skills and self-efficacy to adequately and effectively integrate CS into existing curricula. In this study, 30 pre-collegiate teachers who represent a wide band of experience, grade-levels, and prior CS familiarity participated in a 16-day professional development (PD) course to enhance their content knowledge and self-efficacy in integrating CS into existing lessons and curricula. Using both qualitative and quantitative methodology, a social constructivist approach guided the researchers in the development of the PD, as well as the data collection and analysis on teacher content knowledge and perceptions through a mixed-methods study. Ultimately, participants were introduced to CS concepts and practices through NetLogo, which is a popular multi-agent simulator. The results show that although the pre-collegiate teachers adopted CS instruction, the CS implementation within their curricula was limited to the activities and scope of the PD with few adaptations and minimal systemic change in implementation behaviors.

Keywords: computer science education; computer science; computer science integration; pre-collegiate teacher; K–12 teacher; science education; engineering education

1. Introduction

While technological devices dominate the world today, advanced artificial intelligence (AI) will dominate the day to day functions in the world of tomorrow [1]. The change requires a shift from a technology literate workforce to a highly-skilled workforce knowledgeable in both discipline domains such as science, technology, engineering, and mathematics (STEM), as well as computer science (CS). The good news is that CS is embedded within many workforce STEM careers [2,3]; however, the bad news is that based on current pre-collegiate teacher (those teaching in Kindergarten through 12th grade—K–12—in the United States) CS self-efficacy and skills, CS remains disjointed from many pre-collegiate STEM courses [4,5]. While many informal definitions exist for the exact nature of CS, it can be simply defined as the science of problem solving within a computational context [6]. The distinction between CS, computational thinking (CT), software engineering, and programming is not well defined when only exploring the practical applications rather than the theoretical constructs and underpinnings of the computing spectrum. The distinction is further confounded, as most university-level CS programs prepare software engineers, who utilize broad CT skills, to combine highly specialized CS theory and some specific domain knowledge, to develop software systems through the actionable skill of programming. While most novices might view the entirety of the computing spectrum from the visible tip of the iceberg as programming, the theory and core competencies below the surface form the true basis for a highly skilled CS workforce (Figure 1).

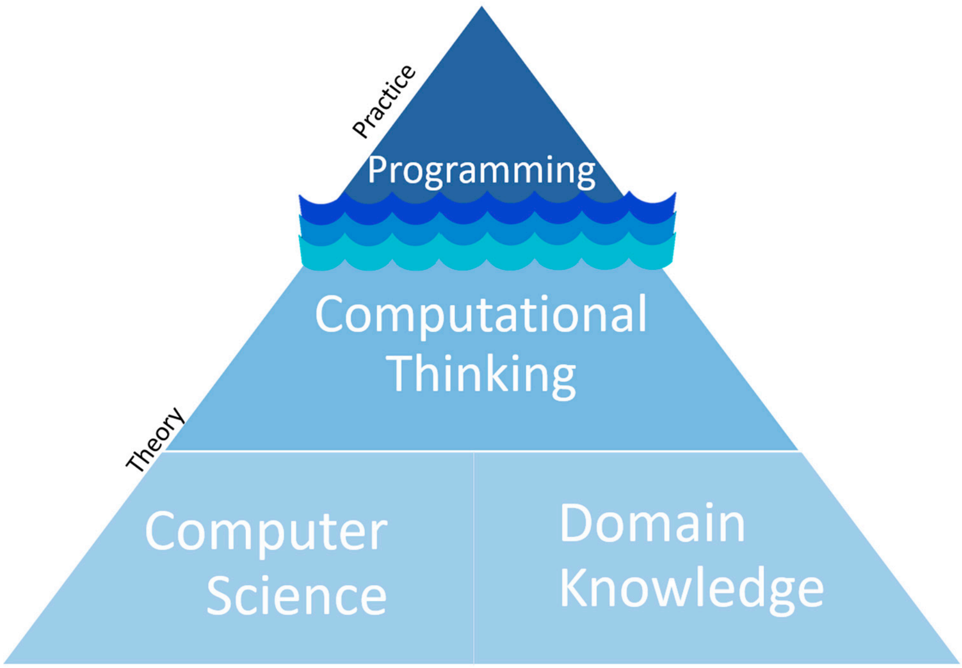

Figure 1. The computing spectrum is shown here as an iceberg model. While the actionable skill of programming is prominently visible to many, the theory and core competencies of computing lie well below the surface.

A more concrete analogy of this model of the computing spectrum surrounds the use of toy construction blocks. Given enough time, one can pick up and learn how to use blocks to create objects (programming), given a tool box of sample construction models (e.g., an arch), one can build more complex structures (computational thinking), and finally if given the foundational theories (domain knowledge) of physics (e.g., forces, material characteristics) and mathematical tools, one can design new custom structures given specific requirements (e.g., computer science and domain knowledge).

In the future, computer scientists and STEM professionals who cannot integrate specific domain knowledge and CS are unlikely to outpace the advances in modern AI and machine learning. For this study, the term 'integration' is key, and integration refers to the blending of CS concepts into already established STEM and other disciplines. Thus, this research study continues to explore the impact of integrating new CS content knowledge into pre-collegiate teachers' prior STEM domain knowledge to produce practical applications through existing pre-collegiate STEM teaching [7]. The incorporation of CS-based problem solving in pre-collegiate classrooms and experiences is substantiated and reinforced through the engineering skills and practices specifically identified in the Next Generation Science Standards (NGSS) [8]. Therefore, pre-collegiate students benefit from the incorporation of CS into their STEM coursework because of the additional exposure to 21st century skills and critical thinking skills, such as those emphasized in the NGSS science and engineering practices (SEP), crosscutting concepts (CCC), and disciplinary core ideas (DCI). Computer science enables these skills such as problem solving, designing solutions, evaluating and analyzing data, peer collaboration, and the oral, written, and electronic dissemination of results.

Currently, CS is taught as a standalone subject in both pre-collegiate and collegiate classrooms. This approach is in sharp contrast to the current use or integration of CS within a variety of STEM fields—from biology, to chemistry, to astronomy, to statistics. The authors argue that pre-collegiate teachers can assist in building a 21st century workforce by incorporating basic CS skills into their established curricula, engaging in effective pedagogy, and experimenting with traditional and cutting-edge resources. The challenge is increasing pre-collegiate teacher self-efficacy within a CS construct without the benefit of a degree in a CS field. The purpose of this study was to investigate a readily accessible online resource, NetLogo, to determine if pre-collegiate teachers' use of NetLogo during professional development (PD) impacts their fundamental CS knowledge, skills, and subject integration.

In this paper, modeling refers to the creation of abstract representations in code. NetLogo is a multi-agent simulator that uses the Logo programming language and was designed for pre-collegiate classroom modeling [9]. The quintessential "Hello World" model for NetLogo consists of modeling ant behavior and pheromone release with food sources [10]. The "Ants" NetLogo model consists of modeling one type of agent, an ant, to move randomly until it detects a chemical 'scent' and then move toward higher concentrations of the scent. Furthermore, an ant releases this chemical scent when it has found food; thus, as more ants locate the scent trail, and thus food, the scent trail itself is reinforced. The inclusion of a free modeling and simulation programming language and environment (web or computer-based), NetLogo [11], offered the pre-collegiate teachers in the PD the opportunity to prepare and develop the skills to incorporate CS concepts into STEM activities. The authors of this study strived to teach the pre-collegiate teachers and pre-collegiate students to think like computer scientists, engineers, and engineering educators to promote modeling.

2. Purpose, Problem, and Research Question

Computer science is in the spotlight of current United States' (U.S.) education policy [12] and the media. While it is recognized that more working CS professionals are needed in the U.S., the path on how to motivate pre-collegiate students into CS majors and careers remains unclear [13]. Currently, most pre-collegiate CS teachers have a collegiate background involving varying degrees of CS, and this process is not scalable to reach all pre-collegiate students. In order to reach as many pre-collegiate teachers as possible, accessible opportunities such as teacher PD need to be offered [14]. The authors of this study address a challenge that today's pre-collegiate teachers face in implementing CS concepts into existing curricula by creating a PD that included: (1) integrating CS into current instruction; (2) explicitly defining real-world CS examples; and (3) showcasing core CS concepts for content knowledge gains. If pre-collegiate teachers possess ample CS content knowledge and high self-efficacy, from PDs or other resources, then they are more likely to incorporate CS into their curricula [6]. This study showcases how pre-collegiate teachers engaged with CS and NetLogo over a 16-day PD called RAMPED, which stands of Robotics, Applied Mathematics, Physics, and Engineering Design. Following the recommendations of other researchers [15], the authors of this study examined: How can pre-collegiate STEM teachers, who have limited CS or programming knowledge, incorporate CS concepts for their STEM classrooms? Due to the research team's interest in the use of the PD material, the central research question of interest evolved to become: ***"How do pre-collegiate STEM teachers view their CS skill set before and after the PD, and do their perceptions align with what they know and how they plan to use CS in their classrooms?"***

3. Theoretical Framework and Literature Review

The authors embraced a social constructivism theoretical framework, where interactions between people (in this case the pre-collegiate teachers) allowed for the creation of connections, and content understanding of the CS material presented and assisted in developing CS self-efficacy and perceptions [16]. NetLogo, as presented here, was based on the group construction of NetLogo ideas, code meanings and changes, and simulated modeling experiences. Teachers created pre-collegiate classroom ideas for CS and NetLogo; their collaborations were collected as evidence. Additionally, the authors utilized Pea and Collins' concept of the fourth wave of science education reform [17] which:

> ... involves the emergence of a systemic approach to designing learning environments for advancing coherent understanding of science subject matter by learners. Science educators and researchers have recognized the need for [mindful] coordination of curriculum

design, activities, and tools to support different teaching methods that can foster students' expertise in linking and connecting disparate ideas concerning science, embedded learning assessments that can guide instructional practices, and teacher professional development supports that can foster continued learning about how to improve teaching practice.

Using technology (e.g., NetLogo programming language) with pre-collegiate teachers, so that they explored and created using CS concepts in K–12 classrooms, was an extension of prior research. Computer science is explored in the following literature review sections related to four main areas including: (1) NetLogo and CS background; (2) pre-collegiate students using and learning CS; (3) pre-collegiate teachers using and learning CS; and (4) higher education students and faculty using and learning CS. These four themes are highlighted in Section 3.2 and add context to the authors' work.

3.1. Background of the Study

For context to the PD, in the following paragraphs, the authors outline what the pre-collegiate teachers investigated for the NetLogo session, and how the material relates to other subjects. The two-day NetLogo PD session was held during the intensive two-week summer PD that was followed by PD support days during the academic year (for a total of 16 days of PD with three dedicated to NetLogo). Pre-collegiate teachers investigated the relationship between common science and manufacturing processes and the design of algorithms to solve optimization and design problems that have no apparent brute-force solution. Of specific focus were investigations of: (1) the biomimicry in genetic algorithms (biological concepts within genetics of population diversity, selection, mutations, and termination) and (2) the relationship between the physical properties of systems (such as the heating and cooling of metals and the balancing of interconnected spring networks) to the creation of megalithic and nanoscopic structures.

The foundations of genetics rely on the intersection of biology, math, and chemistry. During the first part of the teacher-centric investigations on how genetics influences the design of mathematical algorithms, the session built up the baseline knowledge for the pre-collegiate teachers, relying first on the existing knowledge of the pre-collegiate teachers, and then scaffolding and extending the explanation of new concepts by domain content experts. Pre-collegiate teachers then engaged in hands-on, minds-on active learning sessions, with a genetic algorithm that created valid mathematical and chemical equations. After the hands-on/minds-on approach, teachers developed their skill set in either the Python or Sketch programming languages (based on their students' age/skill sets and personal self-selection). Finally, pre-collegiate teachers modified a genetic algorithm template (in Python or Sketch) to solve a simple "game of life" that required setting parameters of birth rate, death rate, and food densities for multiple populations to achieve the maximal survival of a targeted species.

After exposure to the use of genetic-based algorithms, the pre-collegiate teachers investigated algorithms rooted in physics and chemistry. A similar pattern of using the groups' prior knowledge enabled a more realistic starting point for the domain experts to scaffold and improve teachers' understanding. One primary focus of discussion was Hooke's Law (springs) and annealing (forming/breaking crystal lattice structures). Again, the group of teachers chose/investigated two separate algorithms that solved the same problem of where to optimally place human settlements within a geographic area. The session continued by supplementing the prior day's experiences in Python or Sketch, and culminated in the teacher's modification of the "settlement code" to include more realistic constraints, and then compared the results to actual geological settlements.

During the follow-up day session during the academic year, the pre-collegiate teachers were exposed to applications and models of systems that were derived from naturally existing phenomenon, which is an area of research generally encapsulated by "biomorphic systems," "bio-inspired systems," or "biologically derived systems." This one-day session focused more on the physical structures of elements (wings, nests, animal skin coloration) as well on how organisms form collectives based on their fundamental characteristics and interactions with their environment (e.g., flocking birds, schooling fish, ant-food gathering). The ideas of population genetics, survival, and cost, were explored in conjunction with these observable phenomena.

Overall, previous research [18] has indicated "involvement with modeling scientific phenomena and complex systems can play a powerful role in science learning" (p. 151). There have been successes in advancing engineering education and CS through modeling, science standards, and more; however, there is room for improvement in terms of motivating pre-collegiate teachers to use engineering and CS in pre-collegiate classrooms. The study's PD aimed to improve CS content knowledge and motivate pre-collegiate teachers to integrate CS into their STEM disciplines.

3.2. Modeling through NetLogo

In this section, the four literature review themes relate to teaching teachers to think like engineers, and are important for CS and engineering educators at all levels to consider. The first theme of significance to CS and engineering educators focuses on background information about the descriptions of NetLogo as a multi-agent programming language and modeling environment [11,19–22]. NetLogo is a multi-agent simulator that leverages the popular Logo programming language, which was originally developed as a 'learning language.' Multi-agent simulators define the characteristics and/or behaviors of a specific agent (e.g., ant, worker ant, queen ant). Then, the simulator allows an end user to create many replicas of that agent in a predefined world (e.g., ant colony, a flock of birds, atoms, photons). Lastly, the simulator controls and records the interactions of the agents within the world according to the predefined (programmed) rules (e.g., ants following pheromone trails, birds flocking, atoms binding, the behavior of light) [23,24]. Note that an educational use of modeling (e.g., I do, we do, you do) is different than the scientific use of modeling (creation of abstract representation) utilized in this study.

3.2.1. Pre-Collegiate Student Interactions with CS

Secondly, another major literature theme explores pre-collegiate student technology use with CS interactions [15,25–28]. Ultimately, there are a plethora of CS projects for researchers to explore with pre-collegiate teachers and students by using NetLogo or other technologies such as Arduinos and Raspberry Pis, (which the pre-collegiate teachers explored in other RAMPED PD sessions). Although there are several experiential opportunities showcased in the literature, pre-collegiate teachers still struggle to incorporate authentic science, engineering, and CS into their established classroom subjects for their students [29,30].

3.2.2. Pre-Collegiate Teacher Interactions with CS

In the third theme, educational researchers show extensive examples of pre-collegiate teachers using CS in curriculum and instruction, but it is usually in a focused and narrow manner of CS content delivery [9,15,25,31–39]. Thus, based on these works, researchers know that pre-collegiate teachers are attempting and are sometimes successful at incorporating CS into their classrooms. The pre-collegiate teachers' attempted use of CS speaks to the Task Force on Cyberlearning [40] as they call for "research to establish successful ways of using ... technologies to enhance educational opportunities and strengthen proven methods of learning" (p. 7). This is where a PD [or similar program] fits into assisting pre-collegiate teachers with CS in classrooms. Additionally, educational researchers have made arguments that "the cognitive and sociocultural factors related to learning complex systems knowledge are relevant and challenging areas for learning sciences research" [41] (p. 11). Thus, teaching pre-collegiate teachers to utilize CS exploration is complicated, and should be systemic and studied rigorously.

3.2.3. Higher Education Interactions with CS

Finally, although an extension of pre-collegiate teaching, CS is still relevant and needed in higher education, and is encouraged by researchers as well [9,18,31,34,35,38,42–47]. However, as Blikstein and Wilensky [38] point out:

> A common element in those [higher education] programs is to introduce courses in which students design products and solutions for real-world problems, engaging in actual engineering projects. These initiatives have [been] met with some success and are proliferating into many engineering schools. Despite their success, they have not addressed one key issue in transforming engineering education: extending the pedagogical and motivational advantages of design-based courses to theory-based engineering courses, which constitute the majority of the coursework in a typical engineering degree, and in which traditional pedagogical approaches are still predominant (p. 17).

Hence, higher education instructor content and pedagogy and the translation of those elements to pre-collegiate teachers is an area in need of examination and additional study. Computer science can bridge the design to theory issue [4,6,12].

Overall based on the current literature detailed in Section 3.2 there is a clear need for CS instruction, and work in this area is ongoing. Additionally, researchers are now looking beyond the use of CS, and call for an integrated STEM approach [48]. Thus, the need for CS in society, CS in pre-collegiate classrooms, and the call for an integrated STEM approach are the basis for this study.

4. Materials and Methods

With this PD context, experts conducted six independent sessions in two-day blocks during RAMPED, which was a 16-day, year-long engineering education PD focused on CS applications. Nearly two dozen pre-collegiate teachers (n = 22) from a subset of the 30 total STEM teachers participated in the NetLogo PD sessions. In this group of pre-collegiate teachers, several teachers represented a program, SWARMS (Sustaining Wyoming's Advancing Reach in Mathematics and Science), that supports STEM teaching certification with collegiate funding. Although not the focus of the study, the SWARMS teachers were beginning teachers who needed the additional support of this type of CS PD, although they were technologically savvy. The NetLogo session differed from the other five sessions in that it was taught through learner-centric, inquiry-based activities rather than traditional lectures. Remembering that the research question was how pre-collegiate STEM teachers perceive their CS skill set ***before and after the PD, and if the perceptions align with what they know and how they plan to use CS in their classrooms***, the research team collected pre and post-CS content knowledge and self-efficacy data via surveys (including open-ended questions), informal interviews, and artifacts. The research team included faculty from education, CS, physics/astronomy, and engineering. Additionally, an independent evaluator collected qualitative data on PD satisfaction and classroom implementation planning along with quantitative pre and post-content competency data to complement the perception data.

4.1. The Study and Participants

The data for this study were collected intensively during the course of the 2016 16-day engineering education PD for pre-collegiate STEM teachers, and were aimed at enhancing their CS content knowledge and self-efficacy. Of the 22 pre-collegiate NetLogo teacher participants (from a total of 30 STEM pre-collegiate teachers), complete data sets exist for only 20 teachers. The participants equally represented elementary, middle school, and high school teachers in the study. Seven of the pre-collegiate teachers were male, and the other 15 were female. The authors refer to the pre-collegiate teachers as STEM teachers as all taught science or mathematics, and additionally, five of the teachers also taught art, engineering, or technology. The PD focused on CS real-world applications, and the research team implemented six individual PD sessions. Each PD session consisted of two days, which were focused on core CS concepts embedded within authentic uses of CS. The two-day PD sessions included: (1) NetLogo Naturally Inspired (NNI), (2) Astronomy and Space, (3) Robotics, (4) Virtual Reality, (5) Arduinos, and (6) Raspberry Pi. For context, the pre-collegiate teachers chose four of the six sessions to attend during the initial 10-day PD. The pre-collegiate teachers then attended all six extension sessions during the academic year, for a total of 16 participation days. Each session was led by a content expert, with additional guidance and material resources provided by education experts.

The NetLogo session, in comparison to the five other sessions (baseline), was taught through a guided inquiry approach rather than traditional lecture. For example, at the beginning of the first day of the session, participants were presented with a challenge using one of the "Hello World" programs for NetLogo, "Ants" [10], which was described earlier. What follows is a condensed version of those challenges: (1) change the color of the ants, (2) increase the number of food piles that the ants can choose from, (3) introduce the concepts of energy (and death) into the system, (4) introduce reproduction into the system, and (5) introduce some population mutation into the system. These five challenges quickly allowed participants to become acquainted with the language syntax, forced them to utilize fundamental CS techniques to reason and solve the problems, and finally think about the challenges and problems that CS solves that programming alone cannot. This preceded the connections described in Section 3.1.

This study focused on 22 pre-collegiate teachers' CS content knowledge changes, CS self-efficacy, and planned classroom implementation of CS. Pre-collegiate teachers that completed the data set (n = 20) had the following general characteristics: 65% were elementary school teachers (teaching children between five and 14 years old), 63% were science-focused teachers, they had an average of 12.9 years teaching experience, and taught 125.5 students per year, with about 12% of those students on individualized education plans (IEPs).

In order to improve pre-collegiate teacher self-efficacy and CS content knowledge, the RAMPED PD introduced pre-collegiate STEM teachers to real-world applications to CS. Thus, a formal assessment instrument consisted of questions spanning CS applications and fundamental CS theory. The research team created the content questions and administered the pre-assessment and post-assessment instrument to assess pre-collegiate STEM teacher CS knowledge and self-efficacy. This particular study only utilizes a subset of the assessment questions (see Table 1 for the subset of questions) to form a targeted assessment to compare and contrast the NetLogo sessions to other PD sessions. Table 1 highlights seven questions along with the CS concept(s) that they address. Six of the seven questions were related to actual code statements in one of three programming languages: C++, Python, or LOGO, while question seven (Q7), regarding the illustration of sequential operation, only contained graphical illustrations.

Table 1. Subset of assessment questions and corresponding computer science (CS) concept(s) from the full professional development (PD) survey. NNI: NetLogo Naturally Inspired.

Question	Session	Assessment Question (Summary)	CS Concept(s)
Q5	Baseline	Which command queries a robot's joint state?	Syntax
Q7	Baseline	Which of these illustrates sequential operation?	Control Structures
Q14	Baseline	Print out the numbers 1–10	Variables
Q17a	NNI	Show 1000 rolls of a fair 21-sided die	Control Structures
Q17b	NNI	Create a process to swap to two numbers	Syntax
Q17c	NNI	Take a number and add one to it	Variables
Q17d	NNI	Report if a number is even or odd	Boolean Logic

Overall, data were collected several times throughout the study. The participants in the PD answered these previously stated seven survey questions prior to the start of the summer PD session weeks (pre-pre), prior to the specific two-day PD session (pre), immediately after the two-day PD session (post), and immediately after the summer PD session (post-post). These four data sets form the basis for the short-term and long-term impact of the CS content knowledge gains among the pre-collegiate STEM teachers. Thus, each teacher answered the survey questions four times, and the research team collected these data along with other data sets. Interview and artifact data (e.g., lesson plans) were also collected. It is important to note that the authors of this study are not promoting a certain PD or curricula, but instead are promoting a certain mindset in the creation of any PD where there is an emphasis on integration, modeling, and CS concepts.

4.2. Limitations

There are several limitations of the study. Frist, the participant pool is limited to a self-selected group of 20 participants who were admitted on a first-come, first-served basis for a paid PD opportunity, and they chose to complete all of the pre-test and post-test data. Second, the group of participants came largely from the same region and state. Third, the PD was short in terms of teaching a new technical content area to novice pre-collegiate teachers, although the total time spent (120 h) was well above the traditional PD threshold. Fourth, the results in this paper are not generalized due to a limited participant sample, plus an exclusive focus on CS in real-world specific applications. Fifth, the implementation survey was administered three months after the PD concluded; however, the results are not finalized, as the pre-collegiate teachers continue to slowly adopt CS classroom strategies. Sixth, the assessment instrument was custom tailored for the RAMPED PD, and as such had limited reliability and validity. Finally, all of the research team, including the STEM faculty, had significant prior experience in outreach and PD for non-technical audiences; this may be atypical in other technically focused PDs, and may influence the approach of the team, and thus may have skewed the reported results in this study.

5. Analysis and Results

5.1. Qualitative Results

The pre-collegiate teachers answered two types of qualitative data collection, including informal interview questions regarding the individual PD sessions (during working lunches), as well as open-ended questions on the full PD survey. The PD team looked for evidence of content knowledge and perceptions of CS integration. A synthesis of responses shows that the pre-collegiate STEM teachers planned to implement NetLogo into their classrooms at higher rates than the baseline of the other five PD sessions. The research team's summary of the teacher responses from the open-ended questions is summarized in Table 2. Two themes that were identified from their responses included: the cost of the activity implementation and the planned activity type (e.g., inquiry-based, after school club). Also shown in Table 2 are the results of the three-month post-summer PD survey where pre-collegiate teachers reported on both their planned (potential) and already executed (current) CS classroom implementation. The pre-collegiate teachers shared the CS activities that they had planned to use or had already piloted originating from the RAMPED PD experiences (see Table 2).

Table 2. Synthesized aggregation of teaching implementation plan, the current and potential implementation rate, cost, the planned activity, and the PD session type (Inquiry, Explanation, Lecture).

	Topic.	Current (Potential) Implementation	Cost (USD)	Planned Activity	PD Session
	NetLogo	10% (45%)	Free	Inquiry, Family Science Night	Inq. & Exp.
Baseline	Arduino	5% (20%)	<$50	Electricity Unit	Lec. & Inq.
	Raspberry Pi	5% (30%)	<$50	Integrated Project; Afterschool Club	Lec. & Inq.
	Space	0% (10%)	Free	*Unknown*	Lec. & Inq.
	Virtual Reality	5% (25%)	$25–100+	Lecture and Lab	Lec. & Inq.

Along with these implementation data, the following are examples of comments from the pre-collegiate STEM teacher informal interviews and open-ended questions regarding their perceptions of CS integration into STEM:

- [I need] a little more on why a swap is so important in NetLogo or any programming language
- I love the idea of using Net Logo for modeling scenarios with students
- Love the program [NetLogo], my familiarity improved substantially through the workshop, but also with the practice and working with [colleagues]
- [I will use the] applicable web-based opportunities
- My coding background is weak, so I had a hard time figuring out how to modify the code, including for the simulations. However, exposure to the simulations was excellent, so hopefully I can [identify] some of them and use them in class.
- [Doing this workshop has] re-establish[ed] the possibility of using one or more of the simulations models. A benefit would be initial student exposure to coding and using the library of simulations and manipulating the variables already coded.
- [I enjoyed] reviewing the web model and trying to change it.
- The hands-on pieces and the sequence cards were helpful in reading code in a clearer manner.

The research team identified positive terminology regarding participant engagement (e.g., love; excellent; use them; re-established; enjoyed) as well as terms expressing challenges (e.g., a little more; hard time figuring out how; weak) in their responses to NetLogo classroom use. Before PD, the majority of the pre-collegiate teachers (90%) did not know about NetLogo and the potential CS applications. During and after the NetLogo session, the pre-collegiate teachers entertained the idea of integrating CS concepts into their STEM courses.

In actual classroom implementation, one of the elementary teachers used the NetLogo's model's library simulation entitled 'Ants' [10] with her fourth-grade class. The teacher reported that the students changed the size of the ants, the color of the ants, and added a patch of food for the ants by adjusting the NetLogo code. This 'Ant' NetLogo lesson complemented the teacher's life science unit through focusing on biological evolution's unity and diversity. From the NGSS disciplinary core idea on biological evolution 3-LS4-3 [8], the students constructed an argument with evidence that in a particular habitat, some organisms can survive well, some survive less well, and some cannot survive at all. In early 2019, almost three years after the RAMPED PD, this same teacher explained:

> "I have been working with NetLogo in my fourth-grade classroom going on three years now. I began using the online platform in my class after learning about it during a summer PD. In the summer program I rewrote pieces of code in an existing program on erosion to tailor it for fourth-grade science standards. I also created a pre and post-assessment to go with the model to gain data on student learning. The model went well the first year, and students were able to use the model that I had created to see how the flow rate impacted erosion over time. I also used the idea of if/then statements to have my students write a flow chart for the standard subtraction algorithm. It was the first time that I saw students really understand what an algorithm was and why it worked.
>
> When I went to use my model in the second year, NetLogo had changed the program to where my model was no longer operable in the new program, and I had not saved it to the online commons. I unfortunately did not have the time to recode the program, and so I looked for other ways to use NetLogo. My district had also switched to Chrome books, and so the only platform available was online instead of the desktop application. I decided to use NetLogo as a space to teach some simple coding, since our school at that time did not have a coding program. NetLogo was a great way to get students into coding since it is color-coded and is more simple than other coding languages. Students were able to dive in and change the color, size, and shape of the 'turtles' without much experience. I also had the students

simply explore the online library to see what other models were out there. I had students pick a model and talk about what the model showed and how it could be manipulated.

[With the] Hour of Code curriculum, and I no longer felt the need to do coding with NetLogo in my classroom. Students were instead using Scratch and other coding programs to create animations and explore. I am starting to look at NetLogo again with an eye toward physics and modeling collisions of objects to help my students better grasp what is happening, since we do not have the instrumentation to measure the energy change in a collision, and the change is not always discernable to the naked fourth-grade eye. I have found that NetLogo gives me a place to allow my students to model with numerous repetitions, and is highly effective as a supplement to my science instruction."

Also in early 2019, again almost three years after the RAMPED PD, another teacher provided the following vignette:

"When I was the [high school] psychology teacher, a [science teacher] and I (behavioral science) used NetLogo to integrate CS into two problems that had overlap: how a virus spreads and how a social meme spreads across a social network. Students were broken into two groups to look at corresponding NetLogo models and given some basic introduction to coding through these models. In pairs, they made predictions, changed variables, and then tried to make suggestions for how both models could improve. The context of my segment about how a social meme spreads across a network were related to cultural change and cognitive bias (in particular, confirmation bias/motivated reasoning), and the attempts to address the culture problem endemic in [the state] of toxic peer victimization (bullying, peer cruelty, social/relational aggression, cyberbullying) and a local project. It helped students to visualize content [they had] just become familiar with when we jig-sawed Gladwell's Tipping Point in understanding the role of surveillance in identifying mavens, connectors, and salespeople/persuaders that could be enlisted in the effort, and the design of 'sticky' memes to help cause a cultural change or shift. Unfortunately, due to budget cuts, I was reassigned to teach financial literacy, and was not able to continue addressing the cultural problems in [the state]. The applicability of NetLogo also assisted [the science teacher] and [nursing teacher] in the common project that we began the first year [while] helping students understand virology and epidemiology to introduce discussions of vaccination and infection control for CNAs [or certified nursing assistants]."

5.2. Quantitative Results

5.2.1. CS Content Scores

In addition to the CS integration perception data, 20 PD participants responded to all of the content knowledge survey questions on four separate occasions. As discussed in the previous section, the four data collection points of interest in this study focus on the time immediately surrounding the PD, with a pre-summer PD assessment (pre-pre), a pre-PD session assessment (pre), a post-PD session assessment (post), and a post-summer PD assessment (post-post). The average correct score results from a seven-question subset across all 22 participants is shown in Table 3. The trend between the first three assessments is strictly increasing, and the most fluctuation occurs between the post-session assessment (Post) and the post-summer PD assessment (post-post).

Table 3. Percentage of correct CS content answers for PD participants.

Question.	Pre-Pre	Pre	Post	Post-Post
5	6%	18%	44%	26%
7	39%	45%	90%	82%
14	32%	39%	55%	86%
17a	60%	60%	68%	100%
17b	29%	55%	59%	80%
17c	30%	35%	36%	50%
17d	48%	53%	70%	62%

5.2.2. CS—NetLogo Pre-Collegiate Self-Efficacy Scores

The self-efficacy of the pre-collegiate teachers was measured quantitatively on a five-point Likert-scale, with 1 representing "not skillful at all," and 5 representing "extremely skillful." The number of responses to the anonymous self-efficacy survey varied between pre and post, with 22 in pre-PD responses, and 21 post-PD responses. Table 4 shows the overall NetLogo session participant self-efficacy results. When comparing the pre to post results, perhaps most interesting is the overall shift in self-efficacy. Prior to the PD, only 18% (4/22) of pre-collegiate teachers rated themselves at higher NetLogo skill levels (4 or 5); after the PD session 67% (14/21), the same sample rated themselves at higher NetLogo skill levels (4 or 5). This change from four to 14 represents an over 200% increase in the number of pre-collegiate teachers with high self-efficacy after the PD session. When looking at participants rating themselves as average or better (3, 4, or 5), the self-efficacy rate changed from 50% (11/22) to 95% (20/21).

Table 4. Self-efficacy assessment by PD participants pre and post-NetLogo PD session.

Pre (n = 22)	5	6	7	2	2
Post (n = 21)	-	1	6	9	5
Skill Level	1	2	3	4	5
	Not Skillful		*Average*	*Extremely Skillful*	

5.2.3. PD Session Usefulness and Satisfaction

In addition to implementation surveys, content knowledge assessment, and self-efficacy assessments, the PD participants were asked to reflect and provide feedback on the PD itself. Table 5 shows a summary of this anonymous feedback. Immediately after the PD session, during an exit interview, pre-collegiate teachers were asked, "on a three-point scale Likert scale, how useful is NetLogo to you?" Of the 20 respondents, 95% of the pre-collegiate teachers stated that NetLogo was useful to them (moderate extent, 6; larger extent, 13), with only one respondent saying that the session was useful to a negligible extent. The research team asked about the pre-collegiate teacher overall satisfaction, and of the 22 pre-collegiate teachers interviewed after the NetLogo session, over 75% reported being satisfied with the session [1/22 completely dissatisfied (5%), 3/22 mostly dissatisfied (14%), 2/22 moderately satisfied (9%), 5/22 mostly satisfied (23%), and 11/22 completely satisfied (50%)]. Finally, when asked if the NetLogo workshop "stretched teacher thinking into their classrooms", over 85% of teachers believed that it did to a moderate or large extent [8/22 moderate extent (36%), and 12/22 large extent (55%)]. Overall, teachers enjoyed and planned to use integrated CS and NetLogo concepts in their pre-collegiate classrooms after the PD, and expressed that this was in part because their perceived level of CS and NetLogo expertise had increased.

Table 5. Participant assessment of session usefulness and impact.

Session Usefulness (n = 20)	1	0	6	0	13
Session Satisfaction (n = 22)	1	3	2	5	11
Stretched Thinking (n = 22)	2	0	8	0	12
	Low		*Moderate*		*High*

6. Conclusions and Implications

While CS is becoming a "must teach" subject for pre-collegiate teachers, as seen in the literature review, rarely is it incorporated within all STEM disciplines and grade levels. Teachers' lack of CS integration is due in part to a lack of specific CS content knowledge, self-efficacy, and resources to effectively incorporate CS within existing curricula. This limited study shows that using a specific tool or programming language, such as NetLogo, in a PD can create pre-collegiate teacher comfort with the tool along with pre-collegiate teacher classroom use. As seen in Table 3, the pre-collegiate teacher content knowledge scores exhibit a positive increase for all of the questions pre-pre to post, pre-pre to post-post, pre to post, and pre to post-post; this provides support that the NetLogo PD session had a positive impact on short-term (two-week) and long-term (one-year) CS content knowledge. Additionally, most of the pre-collegiate teachers required sustainability and authentic CS integration support, given the 16-day PD with a total of three days with NetLogo immersion. A three-day intensive session can offer beginning CS content knowledge and bolster self-efficacy, but it cannot offer pre-collegiate teachers the in-depth CS knowledge that is need for spontaneous examples during classroom implementation. The participants' CS engagement is encouraging in light of the need for engineering and integrated CS in pre-collegiate classrooms. It is reassuring that overall, the pre-collegiate teachers enjoyed and participated in the NetLogo sessions and planned to use the resource in their classrooms.

The pre-collegiate teacher challenges emphasize the need for pre-collegiate teacher sustained engineering and CS expert support. The teachers identified CS and NetLogo implementation challenges that were traditional in nature (e.g., understanding the programming language) as well as non-traditional (e.g., changing class assignments). In pre-collegiate teacher feedback and artifacts, the authors noticed that there were few real-world examples incorporated with the use of NetLogo in the classroom. This raises concern, as the PD emphasized real-world connections to modeling. The authors speculate that the pre-collegiate teachers need more time to fully understand and internalize the real-world applications of the modeling software as well as more chances for expert collaboration. If the desire to use CS is present, but there are hurdles for pre-collegiate teachers to overcome, then CS instructors from institutes of higher education can assist in filling this gap. As stated earlier, the authors do know about Trautman's blog showcasing the graphic 'Coding Confidence versus Competence,' and admit that this same path might have allowed the pre-collegiate teachers to cling to the 'hand-holding honeymoon' phase of the integration sessions, including the NetLogo experience. However, three years after the PD, two of the NetLogo pre-collegiate teacher participants were utilizing NetLogo in the classroom when possible. This showcases a long-lasting impact from the NetLogo PD, even if it is a small participant pool sample.

The authors argue that pre-collegiate teachers can learn basic CS fundamentals through exploration in a constructivist environment with a free, easily accessible programming language (such as NetLogo) and without structured, lecture-oriented sessions. Potential implications, given a larger focused study, are widespread, as pre-collegiate teachers can potentially increase their own CS content knowledge and self-efficacy. For example, if a teacher participated in free, online modules at their own pace, could that lead to as much of an increase in CS self-efficacy and content knowledge? This could influence integrated CS implementation, which can lead to the incorporation of more CS into pre-collegiate daily activities and standards-based instruction. Finally, the authors believe that STEM teachers, in conjunction with CS content experts, can use NetLogo or a similar technology, as a tool within CS PD. Additionally, the authors propose that pre-collegiate STEM teachers could self-engage or create a NetLogo professional learning community (PLC) to augment these exploratory CS opportunities. Experience with CS and NetLogo appears to increase teacher content knowledge and self-efficacy, and the evidence revealed that pre-collegiate teachers were able to create STEM lessons that incorporated engineering, CS, and NetLogo. Future work could validate studies such as this one and expand STEM teachers' knowledge on what works to incorporate CS into pre-collegiate classrooms. Teaching pre-collegiate teachers to think like engineers and computer scientists is important, timely, and needed. This study shows how integrating CS into existing standards-based curricula can have short and long-term impacts in pre-collegiate classrooms.

Interestingly, although not the focus of this study, five of the six sessions were taught by faculty that used a traditional lecture followed by inquiry experiences, but the NetLogo session was taught by a faculty member who used a brief introduction, and then allowed the pre-collegiate teachers to explore the possibilities of NetLogo modeling in their own exploratory modes (and offered explanations along the way). Could the session approach make the difference in pre-collegiate teacher use of NetLogo? Just as the ants march toward survival in the NetLogo simulation, pre-collegiate teachers are faced with moving forward and integrating CS into STEM disciplines to prepare future generations for the demands and needs of a computing-centric career. Similar to ants finding food, pre-collegiate teachers who adopt and develop integrated CS materials become beacons to other pre-collegiate teachers, enabling them to address the collective need for CS integration throughout the pre-collegiate education system. The need for CS and the standards that call for CS and CS-like skills are present and currently appearing; thus, pre-collegiate teachers need an institute of higher education or other programs' support in varied dimensions and on copious occasions to allow as many pre-collegiate teachers as possible to embrace and utilize integrated CS in classrooms. Based on the data and analysis presented, the authors offer that NetLogo (or any other similar) technology, when used in an exploratory, inquiry-based fashion, is capable of enabling teachers to incorporate CS into their pre-collegiate classrooms especially when pre-collegiate teachers embrace content learning and show increasing perception of their abilities within the CS context.

Author Contributions: Conceptualization, M.B., A.C.B.; methodology, M.B.; validation, M.B., A.C.B.; formal analysis, M.B.; investigation, M.B., A.C.B.; resources, M.B., A.C.B.; data curation, M.B., A.C.B.; writing—original draft, M.B., A.C.B.; writing—revision and editing, A.C.B., M.B.; visualization, M.B., A.C.B.; supervision, M.B., A.C.B.; project administration, A.C.B.; funding acquisition, A.C.B.

Funding: This work was supported by 1) US federal grant, RAMPED, under No Child Left Behind (NCLB) (P.L.107F110, Title II, Part B) administered by the Wyoming Department of Education (MSP Grant #1601506MSPA2); and 2) National Science Foundation (NSF) Noyce—called SWARMS—(NSF Grant# 1339853). Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Acknowledgments: This work is a revised and extended analysis and discussion of data presented in short-paper published by the authors: Burrows, A.C. and Borowczak, M. "Teaching Teachers to Think Like Engineers Using NetLogo," which, appeared in the proceedings of the 2017 ASEE Annual Conference & Exposition, Columbus, OH, USA, 2017. The authors would like to acknowledge all the pre-collegiate teacher participants from the RAMPED PD, and especially Crystal and Joshua for their specific insights and feedback to this work.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Pew Research Center. AI, Robotics, and the Future of Jobs. Available online: http://www.pewinternet.org/2014/08/06/future-of-jobs/ (accessed on 10 December 2018).
2. U.S. Bureau of Labor Statistics. Available online: https://www.bls.gov/ooh/computer-and-information-technology/home.htm (accessed on 10 December 2018).
3. U.S. Bureau of Labor Statistics. Available online: https://www.bls.gov/spotlight/2017/science-technology-engineering-and-mathematics-stem-occupations-past-present-and-future/pdf/science-technology-engineering-and-mathematics-stem-occupations-past-present-and-future.pdf (accessed on 10 December 2018).

4. Hubwieser, P.; Giannakos, M.; Berges, M.; Brinda, T.; Diethelm, I.; Magenheim, J.; Pal, Y.; Jackova, J.; Jasute, E. A Global Snapshot of Computer Science Education in K-12 Schools. In Proceedings of the 2015 ITiCSE on Working Group Reports, Vilnius, Lithuania, 4–8 July 2015; pp. 65–83.
5. Celik, V.; Yesilyurt, E. Attitudes to technology, perceived computer self-efficacy and computer anxiety as predictors of computer supported education. *Comput. Educ.* **2013**, *60*, 148–158. [CrossRef]
6. Burrows, A.C.; Borowczak, M.; Slater, T.F.; Haynes, J.C. Teaching computer science and engineering through robotics: Science and art form. *Probl. Educ. 21st Century* **2012**, *47*, 6–15.
7. Leonard, J.; Buss, A.; Gamboa, R.; Mitchell, M.; Fashola, O.S.; Hubert, T.; Almughyirah, S. Using robotics and game design to enhance Children's self-efficacy, STEM attitudes, and computational thinking skills. *J. Sci. Educ. Technol.* **2016**, *25*, 860–876. [CrossRef]
8. NGSS Lead States. *Next Generation Science Standards: For States by States*; National Academies Press: Washington, DC, USA, 2013.
9. Wilensky, U.; Papert, S. Restructurations: Reformulations of knowledge disciplines through new representational forms. In Proceedings of the Constructionism 2010: Constructionist approaches to Creative Learning, Thinking, and Education: Lessons for the 21st Century, Paris, France, 16–20 August 2010.
10. Wilensky, U. NetLogo Ants Model. Center for Connected Learning and Computer-Based Modeling. Northwestern University: Evanston, IL, USA. Available online: http://ccl.northwestern.edu/netlogo/models/Ants (accessed on 10 December 2018).
11. Wilensky, U.; NetLogo. Center for Connected Learning and Computer-Based Modeling. Northwestern University: Evanston, IL, USA. Available online: http://ccl.northwestern.edu/netlogo (accessed on 10 December 2018).
12. Obama, B. The President's Radio Address: Giving Every Student an Opportunity to Learn Through Computer Science for All. Office of the Press Secretary, The White House. Available online: https://www.whitehouse.gov/the-press-office/2016/01/30/weekly-address-giving-every-student-opportunity-learn-through-computer (accessed on 10 December 2018).
13. Cannady, M.A.; Greenwald, E.; Harris, K.N. Problematizing the STEM Pipeline Metaphor: Is the STEM Pipeline Metaphor Serving Our Students and the STEM Workforce? *Sci. Educ.* **2014**, *98*, 443–460. [CrossRef]
14. Borowczak, M.; Burrows, A.C. Interactive Web Notebooks Using the Cloud to Enable CS in K–16+ Classrooms and PDs. In Proceedings of the 2017 ASEE Annual Conference & Exposition, Columbus, OH, USA, 24–28 June 2017.
15. Sengupta, P.; Dickes, A.; Farris, A.V.; Karan, A.; Martin, D.; Wright, M. Programming in pre-collegiate science classrooms. *Commun. ACM* **2015**, *58*, 33–35.
16. Vygotsky, L. Interaction between learning and development. In *Readings on the Development of Children*; Gauvain, M., Cole, M., Eds.; Scientific American Books: New York, NY, USA, 1978; pp. 34–40.
17. Pea, R.D.; Collins, A. Learning how to do science education: Four waves of reform. In *Designing coherent science education*; Kali, Y., Linn, M.C., Roseman, J.E., Eds.; Teachers College Press: New York, NY, USA, 2008.
18. Hashem, K.; Mioduser, D. The Contribution of Learning by Modeling (LbM) to Students' Understanding of Complexity Concepts. *Int. J. e-Educ. e-Business, e-Manag. e-Learn.* **2011**, *1*, 151.
19. Blikstein, P.; Abrahamson, D.; Wilensky, U. Netlogo: Where we are, where we're going. In Proceedings of the annual meeting of Interaction Design and Children, Boulder, Colorado, 8–10 June 2005.
20. Tisue, S.; Wilensky, U. Netlogo: A simple environment for modeling complexity. In Proceedings of the International Conference on Complex Systems, Boston, MA, USA, 16–21 May 2004; Volume 21, pp. 16–21.
21. Tisue, S.; Wilensky, U. NetLogo: Design and implementation of a multi-agent modeling environment. In Proceedings of the Agent 2004 Conference on Social Dynamics: Interaction, Reflexivity and Emergence, Chicago, IL, USA, 7–9 October 2004; pp. 7–9.
22. Wilensky, U. Modeling nature's emergent patterns with multi-agent languages. In Proceedings of the 8th Conference of EuroLogo, Linz, Austria, 21–25 August 2001; pp. 1–6.
23. Goel, A.K.; Rugaber, S.; Vattam, S. Structure, behavior, and function of complex systems: The structure, behavior, and function modeling language. *Artif. Intell. Eng. Des. Anal. Manuf.* **2009**, *23*, 23–35. [CrossRef]
24. Malan, D.J.; Leitner, H.H. Scratch for budding computer scientists. *ACM SIGCSE Bull.* **2007**, *39*, 223–227. [CrossRef]

25. Basu, S.; Dickes, A.; Kinnebrew, J.S.; Sengupta, P.; Biswas, G. CTSiM: A Computational Thinking Environment for Learning Science through Simulation and Modeling. In Proceedings of the 5th International Conference on Computer Supported Education, Aachen, Germany, 6–8 May 2013; pp. 369–378.
26. Berland, L.K.; Reiser, B.J. Classroom communities' adaptations of the practice of scientific argumentation. *Sci. Educ.* **2011**, *95*, 191–216. [CrossRef]
27. Svihla, V.; Linn, M.C. A design-based approach to fostering understanding of global climate change. *Int. J. Sci. Educ.* **2012**, *34*, 651–676. [CrossRef]
28. Vattam, S.; Goel, A.K.; Rugaber, S.; Hmelo-Silver, C.E.; Jordan, R.; Gray, S.; Sinha, S. Understanding Complex Natural Systems by Articulating Structure-Behavior-Function Models. *Educ. Technol. Soc.* **2011**, *14*, 66–81.
29. Burrows, A.C. Partnerships: A systemic study of two professional developments with university faculty and K–12 teachers of science, technology, engineering, and mathematics. *Probl. Educ. 21st Century* **2015**, *65*, 28–38.
30. Burrows, A.C.; DiPompeo, M.A.; Myers, A.D.; Hickox, R.C.; Borowczak, M.; French, D.A.; Schwortz, A.C. Authentic science experiences: Pre-collegiate science educators' successes and challenges during professional development. *Probl. Educ. 21st Century* **2016**, *70*, 59–73.
31. Blikstein, P. Digital fabrication and 'making' in education: The democratization of invention. *FabLabs Mach. Mak. Invent.* **2013**, *4*, 1–21.
32. Donnelly, D.F.; Linn, M.C.; Ludvigsen, S. Impacts and characteristics of computer-based science inquiry learning environments for precollege students. *Rev. Educ. Res.* **2014**, *84*, 572–608. [CrossRef]
33. Grover, S.; Pea, R. Computational Thinking in K–12 A Review of the State of the Field. *Educ. Res.* **2013**, *42*, 38–43. [CrossRef]
34. Klašnja-Milićević, A.; Vesin, B.; Ivanović, M.; Budimac, Z. E-Learning personalization based on hybrid recommendation strategy and learning style identification. *Comput. Educ.* **2011**, *56*, 885–899. [CrossRef]
35. Maroulis, S.; Guimera, R.; Petry, H.; Stringer, M.J.; Gomez, L.M.; Amaral, L.A.N.; Wilensky, U. Complex systems view of educational policy research. *Science* **2010**, *330*, 38–39. [CrossRef] [PubMed]
36. Pathak, S.A.; Kim, B.; Jacobson, M.J.; Zhang, B. Learning the physics of electricity: A qualitative analysis of collaborative processes involved in productive failure. *Int. J. Comput.-Support. Collab. Learn.* **2011**, *6*, 57–73. [CrossRef]
37. Sengupta, P.; Kinnebrew, J.S.; Basu, S.; Biswas, G.; Clark, D. Integrating computational thinking with pre-collegiate science education using agent-based computation: A theoretical framework. *Educ. Inf. Technol.* **2013**, *18*, 351–380. [CrossRef]
38. Shen, J.; Lei, J.; Chang, H.Y.; Namdar, B. Technology-enhanced, modeling-based instruction (TMBI) in science education. In *Handbook of Research on Educational Communications and Technology*; Springer: New York, NY, USA, 2014; pp. 529–540.
39. Wilensky, U.; Brady, C.E.; Horn, M.S. Fostering computational literacy in science classrooms. *Commun. ACM* **2014**, *57*, 24–28. [CrossRef]
40. Borgman, C.L.; Abelson, H.; Dirks, L.; Johnson, R.; Koedinger, K.R.; Linn, M.C.; Lynch, C.A.; Oblinger, D.G.P.; Roy, D.S.; Katie Smith, M.S.; et al. *Fostering Learning in the Networked World: The Cyberlearning Opportunity and Challenge. A 21st Century Agenda for the National Science Foundation*; Report of the NSF Task Force on Cyberlearning; DIANE Publishing Company: Collingdale, PA, USA, 2008.
41. Jacobson, M.J.; Wilensky, U. Complex systems in education: Scientific and educational importance and implications for the learning sciences. *J. Learn. Sci.* **2006**, *15*, 11–34. [CrossRef]
42. Blikstein, P. Using learning analytics to assess students' behavior in open-ended programming tasks. In Proceedings of the 1st International Conference on Learning Analytics and Knowledge, Banff, AB, Canada, 27 February–1 March 2011.
43. Blikstein, P.; Wilensky, U. Less is more: Agent-based simulation as a powerful learning tool in materials science. In Proceedings of the IV International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005), Utrecht, Holland, The Netherlands, 25–29 July 2005.
44. Blikstein, P.; Wilensky, U. *MaterialSim: A constructionist agent-based modeling approach to engineering education. Designs for Learning Environments of the Future*; Springer: Boston, MA, USA, 2010; pp. 17–60.
45. Chiu, M.H.; Wu, H.K. The roles of multimedia in the teaching and learning of the triplet relationship in chemistry. In *Multiple Representations in Chemical Education*; Springer: Netherlands, The Netherlands, 2009; pp. 251–283.

46. Levy, S.T.; Wilensky, U. Crossing levels and representations: The Connected Chemistry (CC1) curriculum. *J. Sci. Educ. Technol.* **2009**, *18*, 224–242.
47. Sengupta, P.; Wilensky, U. Learning electricity with NIELS: Thinking with electrons and thinking in levels. *Int. J. Comput. Math. Learn.* **2009**, *14*, 21–50. [CrossRef]
48. Burrows, A.C.; Garofalo, J.; Barbato, S.; Christensen, R.; Grant, M.; Parrish, J.; Thomas, C.; Tyler-Wood, T. Integrated STEM and Current Directions in the STEM Community. *Contemp. Issues Technol. Teach. Educ.* **2017**, *17*, 478–482.