The Pocketworld Playground: Engaging Online, Out-of-School Learners with Agent-based Programming

John Chen
Northwestern University
Evanston, IL, United States of America
civitas@u.northwestern.edu

Lexie Zhao
Northwestern University
Evanston, IL, United States of America
xinyuezhao2020@u.northwestern.edu

Michael S. Horn
Northwestern University
Evanston, IL, United States of America
michael-horn@northwestern.edu

Uri J. Wilensky
Northwestern University
Evanston, IL, United States of America
uri@northwestern.edu

ABSTRACT

Agent-based modeling (ABM) has become a major approach to promote computational thinking and complex systems thinking in K-12 education. However, agent-based programming (ABP), the computational foundation of ABM, is less defined and discussed in previous literature. Summarizing previous studies around ABP from computer science and education, we argued for the potential benefits of introducing ABP to youth. Rooted in the interest development theory, we presented the design of a scaffolded agent-based programming space, the Pocketworld Playground (POP), that aims to engage out-of-school online young learners through developing their interest in ABP. The POP was built in Turtle Universe (TU), the mobile incarnation of NetLogo. Using a mixed-methods approach to analyze log data and artifacts created by learners, we found that POP successfully engaged learners with ABP practices; helped develop situational and individual interest; and contributed to TU's emerging online community. Finally, we discussed design lessons that could benefit other online learning designers.

CCS CONCEPTS

• **Human-centered computing → Collaborative and social computing systems and tools**; **Empirical studies in collaborative and social computing**; **Empirical studies in ubiquitous and mobile computing**.

KEYWORDS

Agent-based Programming; Agent-based modeling; Computational Literacy; Informal Learning; Constructionism

ACM Reference Format:
John Chen, Lexie Zhao, Michael S. Horn, and Uri J. Wilensky. 2023. The Pocketworld Playground: Engaging Online, Out-of-School Learners with Agent-based Programming. In *Interaction Design and Children (IDC '23), June 19–23, 2023, Chicago, IL, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3585088.3589357

IDC '23, June 19–23, 2023, Chicago, IL, USA

ACM ISBN 979-8-4007-0131-3/23/06…$15.00
https://doi.org/10.1145/3585088.3589357

1 INTRODUCTION

Computational thinking is a critical form of literacy that supports a widening list of social and productive activities (e.g., [61, 72]). There have been significant efforts to promote computational thinking and participation in K-12 education (e.g. [36]). One popular approach is Agent-based modeling (ABM), which promotes computational thinking and complex systems perspectives [1, 55, 65]. Originating from complex systems theory, ABM is a powerful methodology that leverages individual autonomous agents and simple computational rules to study complex emergent phenomena in natural and social science [24, 31, 64]. ABM promotes a decentralized, probabilistic mindset [66] and opens up opportunities to bring a new representational infrastructure that can be used as a restructuration [61–63] of scientific knowledge.

Agent-based programming (ABP) is the technical foundation of Agent-based Models (ABMs). Inspired by literature in computer science (e.g. [4, 14, 56, 75]) and learning sciences [55], we define ABP as a programming paradigm that focuses on the behaviors of individual agents to produce emergent macro-level patterns. Similar to ABM, ABP is decentralized and often probabilistic in nature. While ABM encompasses multiple activities ranging from creation, validation, and analysis, it relies on ABP to produce desired reference phenomena [64]. When the goal is to have learners change, critique, or create models, the syntactic and semantic ABP skills are particularly essential to create sets of agents, make rules of their individual behaviors, and engage them in interactions with the world and each other [64].

However, integrating ABM in K12 science and math curricula still poses a significant challenge. One reason is that computational modeling and programming involve a higher overhead for teaching and learning in classrooms where teachers and students could be less prepared for CS-related content [55]. Adding to the challenge, ABP brings a different paradigm than most programming taught in schools and involves a decentralized, probabilistic mindset (see the definition in [66]). As a result, many implementations of ABM in K-12 classrooms have been limited to exploration, where students play with models built by others and have limited opportunities

or support to construct their own (e.g. [28, 32, 41]). Without further support for the learning of ABP, it becomes challenging to materialize the full learning potential of ABM.

To address the challenge, one possible solution is to explicitly support the learning of ABP. Over the past decade, several studies have explored ways to support learners' deeper engagement with ABM. Both [34, 55, 71] leveraged block-based visual programming to lower the threshold of ABP in ABM learning activities. In addition, studies have introduced ABP in other creativity-oriented learning activities, such as creative art, games, or expressive programming (e.g. [8, 9, 25]). By creating more inclusive opportunities for learners to engage in relatable agent-based computational activities, these studies have the potential to reach out to a wider audience with ABP and prepare learners for deeper engagement with ABM.

Existing studies on computational thinking education have primarily focused on classroom settings [58], and only a few ABM & P learning activities were implemented outside schools (e.g. [9, 34]). Given that 85% of children's awake time is spent outside classrooms [44], youth's out-of-school computational practices are receiving more attention (e.g., [39]). The growing use of mobile technology among youth [3] brings further opportunities for researchers and practitioners to engage out-of-school learners with computational activities.

To take advantage of this opportunity, Turtle Universe (TU) builds on NetLogo [69], the most widely used "low ceiling, high threshold" ABM environment, and extends the access of ABM & P to smartphones and tablets. The availability of TU on smartphones and tablets makes it possible to invite online, out-of-school learners to voluntarily participate in ABM & P [15, 16]. Whereas, without proper design to bolster learners' short-term and long-term interests, increased opportunities do not naturally lead to increased participation [46].

In this paper, we introduce and investigate the first learning design, the Pocketworld Playground (POP), that seeks to broaden the participation of ABP by developing learners' situational and individual learning interests. Designed for and implemented in Turtle Universe, POP is an agent-based programming environment with guided pathways that specifically target online, out-of-school learners at K-12 age level. Using the Integrated Interest Development for Computing Education Framework (IIDCEF, [46]), we discuss our design decisions that aim to support learners' interests. Specifically, we are interested in the design-based research questions:

- RQ1. Did our learning design support learners' engagement with the ABP paradigm?
- RQ2. Did our learning design support learners' situational (or short-term) interests in ABP?
- RQ3. Did our learning design support learners' individual and collective interests in ABP?

Using log data collected from 14,710 online, out-of-school learners and 934 POP projects shared in the online community, we adopted behavioral metrics proposed by the IIDCEF to answer the RQs. We will then discuss the implications for learning designers to 1) design activities that engage learners with ABP; 2) design environments that support online, out-of-school learners' short-term and long-term interests in computational activities.

2 RELATED WORK

2.1 Arguing for Agent-based Programming (ABP)

In the previous section, we defined Agent-based Programming (ABP) as a programming paradigm that focuses on the behaviors of individual agents to produce emergent macro-level patterns. Reviewing the previous studies from computer science and education, we further found that ABP has the potential to promote decentralized and probabilistic thinking in learning contexts.

In computer science, the definition of ABP focuses on a decentralized paradigm of programming. [56] first proposes the framework of Agent-Oriented Programming in AI, strictly defining agents as entities with "mental components", such as beliefs, capabilities, choices, and commitments. Later work from [4, 49] focuses on agents' centrality and their "behavior, motivations, and relationships" in natural and social science modeling. The definition was further broadened to include general-purpose software, as [75] characterized: 1) Autonomy, that an agent is not passively subject to a global, external flow of control; 2) Situatedness, that an agent performs its actions in a particular environment; and 3) Sociality, that agents may communicate or coordinate with each other to achieve local or global objectives. Recently, [14] starts to distinguish ABP as a distinct construct, with ABM as a type of its application. Instead of writing a program that directly produces the desired macro-level pattern, ABP focuses on the micro-level, operating on the behaviors of individual agents.

In education, the definition of ABP is less clear and often associated with the programming aspect of ABM. However, the probabilistic way of programming is more often foregrounded. Both [23, 45, 55, 74] used the term "Agent-based Programming" to distinguish their ABM activities with programming elements from studies that only involve the usage of agent-based models (ABMs). In cases where ABP was used in educational scenarios other than supporting ABM, most of them involve probabilistic programming as well. For example, ABP is used to 1) produce generative art or "expressive coding" (e.g. [6, 9, 29]); or 2) design serious games (e.g. [48]); or 3) control multi-agent systems, such as swarm robot systems (e.g. [20, 22]). In these examples, probabilistic thinking is often implied (e.g. in swarm robot systems, where sensors and motors are not 100% accurate) or foregrounded (e.g. in generative art, where probabilistic programming could lead to diverse effects).

In learning contexts, it seems that ABP has the potential to promote a decentralized and probabilistic mindset [66]. It prepares learners for future engagement with ABM not only through familiarizing them with programming, but also by helping them develop the necessary mindset for ABM. ABP also provides a wider array of computational activities for learners to identify with and become interested in.

2.2 Engaging Online, Out-of-school Learners with ABP

With the advent of widespread mobile adoption, engaging online, out-of-school learners has become increasingly important. Studies have shown that weaving in-school and out-of-school learning experiences together could maximize learning gains [12]. In addition,

as equity issues in computing education partially originate from unequal access to learning technologies [26, 60], making computational learning activities publicly available becomes an alternative and supplementary pathway to address them. Online communities bring further opportunities for engaging diverse audiences and cultural practices (e.g. [52, 53]).

However, only a few studies try to engage out-of-school learners with ABP (e.g. [9, 34]), and none have focused on online learners. One reason for this is that no ABM or P environments have provided sufficient features for designing and distributing such learning experiences. While Turtle Universe solves both issues by providing a technical infrastructure to design, implement, and distribute interactive learning experiences [18], it poses new challenges for learning designers. How can we design an interactive learning experience for online, out-of-school learners in ABP?

Informed by the Integrated Interest Development for Computing Education Framework (IIDCEF, [46]), we recognize that a good design should not only provide new opportunities for learners to engage with ABP, but also support their interests. Following [50]'s theory, we differentiate interest into two constructs: situational interest, which refers to a short-term, localized interest that could be triggered by a particular learning activity; and individual interest, which refers to a long-term and slowly shifting relationship between learners and the learning activity. Both frameworks emphasize the interaction between short-term situational interest and long-term individual interest. They also underscore the impact of support (or lack thereof) from the learning space on the development of interests and focus on the notion of voluntary re-engagement [46] or continued engagement [50] as the manifestation and potential measurement of interests. Additionally, interests are also highly personal: a learning space may simultaneously boost some learners' interests and decrease others.

Below, we briefly review prior studies around developing interests in computational activities that inspired or informed our design. We organize them around the three key dimensions proposed by the IIDCEF [46]: Knowledge; Value; and Belonging.

2.2.1 Knowledge. The IIDCEF distinguishes three key factors that contribute to meaningful knowledge building that could be reciprocally beneficial for interest development: 1) Appropriate levels of challenge; 2) Personally relevant learning contexts; and 3) Authentic computational learning activities. Started by Papert [47], many studies following the constructionist tradition have addressed these factors. Papert's own Mathland metaphor is a great example: The design of computers should ensure that communicating with them can be a natural process, so that "talking mathematics" could be similar to "learning French while living in France" [47]. NetLogo is designed with the same principles: its grammar is centered around manipulating multiple agents at the individual level [59]. It can be used for multiple purposes, focused mostly on scientific modeling but also enabling expressive arts and gaming, thus supporting many culturally relevant learning activities (e.g. [10, 43]).

However, it is still challenging to implement learning activities with ABP elements within a short period of time, even in classrooms (e.g. [55]). Recently, many block-based programming environments were introduced for ABM & P to further lower the threshold of participation, especially in introductory or informal settings. We categorize them by degrees of generalization (Fig. 1). Domain-specific block-based programming environments are characterized by their small number of blocks and low threshold to learn. For example, young learners in a museum were found to start programming with the Frog Pond [34] within 1-2 minutes. The few blocks in those environments are often designed to represent domain-specific behavior, such as "hop", "chirp" for frogs. Domain-general block-based programming environments provide more expression power through content-agnostic blocks: instead of "hop", learners would use a "forward" block with a "distance" parameter to achieve the same behavior. They are much more powerful, but also come with far more blocks and a higher threshold of learning. For example, StarLogo TNG's teacher guide [2] suggests 10 lessons for learning the foundation of the environment.

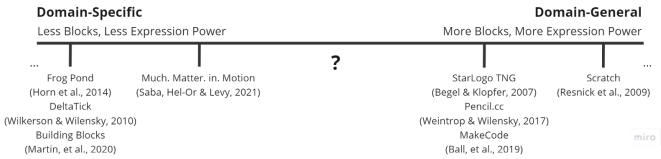

Figure 1: Conceptualization of Block-based Programming Environments

While existing scaffolding efforts for ABM & P usually focus on conceptual learning (e.g. [5]), another direction to support the learning is through interactive scaffolds. Here, we adopt Collins' definition [21] of scaffolding which revolves around experts' support for novices to carry out tasks. Successful scaffolding leads to its own removal: when the scaffolds are faded, learners could still carry out the same tasks [21]. [35] further discussed two design strategies of technology-enabled scaffolds: learner-adaptive, where the technical design will automatically change to respond to learners' needs; and learner-adaptable, where the technical design enables learners to initiate the fading of scaffolds. Since then, the two strategies have been used to support various learning activities and verified for their efficacy (e.g. [57] for adaptive; and [38] for adaptable). The introduction of interactive technologies also presents new opportunities for designing scaffolds with personal relevance (e.g. [42]), with a potential to support value and belonging in learning experiences.

2.2.2 Value and Belonging. The IIDCEF nominates value and belonging as two other critical dimensions for interest development, both highly personal. Value comes from learners' personal perception of usefulness, importance, and feasibility of the learning activity, while the sense of belonging emerges as learners connect with the learning activity they are interested in [46]. To support the development of interest, the IIDCEF also identifies several strategies: 1) provide spaces for learners to create personally meaningful projects and help learners create visions of their work's utility; 2) foreground instruction that aligns with learners' goals; 3) facilitate learner-driven exploration; 4) broaden the scope of computing beyond formal curricula; 5) provide alternative accounts for "who does computing". Many of those strategies could be found in studies that involve online learning communities for youth. For example,

the MOOSE Crossing community engaged thousands of learners with gamified computational activities [13]. Since learners need to type in commands for every move in the game world, they easily found usefulness and importance in programming. The open-ended world design also facilitates learner-center exploration and prompts learners to seek more knowledge of the programming language [13]. Similarly, the Scratch community [51] is full of online, out-of-school learners working on their diverse projects. The connected learners in Scratch formed their collective identities and cultures [11, 27], manifesting a high degree of value and belonging.

3 LEARNING DESIGN

We present the technology-enabled learning design of the Pocketworld Playground (POP, Fig. 2) that focuses on engaging and developing interest for online, out-of-school young learners in ABP. Through the name "Pocketworld Playground", we tried to communicate two layers of meanings: First, the environment is designed as a playground, instead of a playpen [7], for learners to explore and tinker on their own; Second, it is aimed for learners to create a "pocket world" of their own. Since 2021, POP was implemented as a publicly available online learning activity in Turtle Universe.

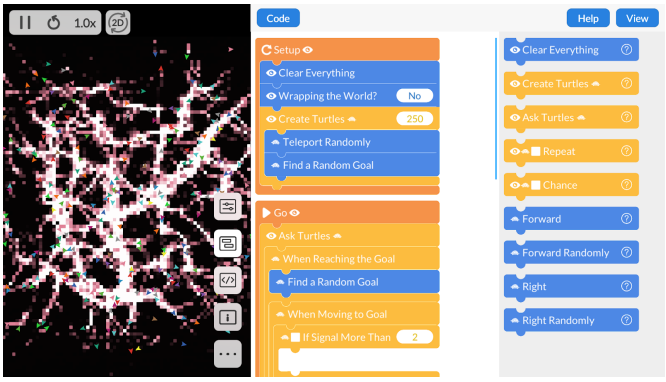

Figure 2: POP's Immersive Programming Environment.

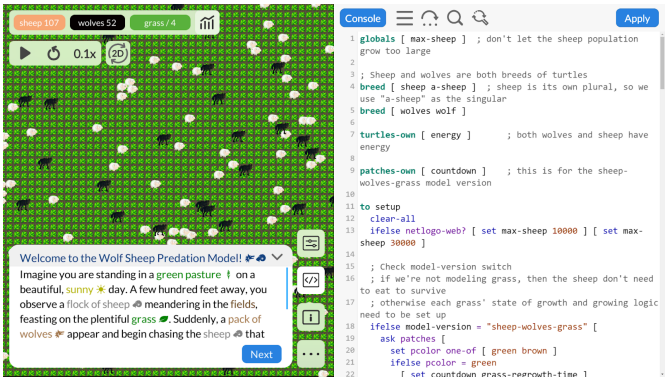

Figure 3: The Wolf Sheep Predation Model in TU

3.1 Turtle Universe

Fig. 3 illustrates the classical Wolf Sheep Predation model [68] presented in Turtle Universe (TU) on a smartphone. On the left are the viewport, plots, and controls of the agent-based model; on the right, a series of tabs allows learners to switch between the widgets, NetLogo code, the introduction, and other actions (e.g. saving, loading, remixing, or leaving the model). With the ability to run natively on tablets and smartphones, TU's mobile-oriented interface brings new opportunities to engage online, out-of-school learners with ABM & P learning activities.

Serving online, out-of-school K-12-aged learners, TU's learning context is inherently different from school contexts. Instead of recruiting students from school or afterschool settings, TU invites participation through App Store and Google Play. TU was also introduced in the Physics Lab AR community [19], another constructionist learning app for youth that focuses on physics-related activities. However, this presents several challenges for learning design. Challenge 1: While many learners might already have participated in constructionist learning experiences (e.g. building their own physics experiment), most of them did not have prior exposure to any ABM & P learning activities. Also, the remote-only informal setting makes it difficult to implement rigid task structures. Challenge 2: The early engagement statistics of the platform show the highly interest-driven nature of learners' participation: for each visit to a model, a learner would only spend 86s on average and the median was 40s. Comparable to museum settings, most visitors have the agency to decide whether, when, and to what extent to engage in learning activities. Depriving users of their agency could easily lead to frustration. Challenge 3: TU was originally designed as a learning environment for ABM. Yet, learners' interests tend to be more diverse, as popular projects in the app often included generative arts and games. Learners also expressed a deep interest in creating new projects. However, as text-based coding poses a higher threshold, only a few learners started to share remixes of models typically with minimal changes.

To engage and develop interests in online, out-of-school young learners in ABP, we designed the POP as a scaffolded agent-based programming space presented in TU that incorporates: 1) An immersive ABP environment, where learners program toward their own goals naturally in a decentralized and probabilistic approach; 2) A set of learner-adaptable and learner-adaptive scaffolds to support learners' emergent needs; and 3) A network of open-ended learning pathways that relates to individuals and the community of learners. The three parts of our design are linked to our research questions and will be examined by our empirical study.

3.2 An Immersive ABP Environment

The POP environment has two main layers: the block-based environment (Fig. 3), where learners leverage programming blocks to create their own projects; and the interactive scaffolds and pathways to support them. Our first goal is to design an ABP environment that lowers the threshold for first-time learners while still providing the potential for "high-ceiling" projects. Here, our definition of "immersion" is different from its typical usage: inspired by Papert's MathLand metaphor [47], we aim to create an immersive "ABP-land" where the native way of programming is agent-based.

> **Design Decision 1.** We designed the programming environment of the POP to be domain-plural, where the natural way of programming is decentralized and non-deterministic.

To achieve our design goal, previous efforts have often pointed to domain-specific block-based programming environments (Fig. 1). While the technological infrastructure is content-neutral (e.g. DeltaTick, [71]; NetTango Web, [33], their implementations are often dedicated to singular topics: a frog pond in the Frog Pond [34], flu-like disease spread [73], etc. While learners could program to achieve a multitude of end results, they are often bound to a fixed context that is pre-designed according to pedagogical needs. Domain-general environments, on the contrary, provide high flexibility at the cost of learning thresholds.

Reflecting on existing types of programming environments, our design aims to support a variety of ABP opportunities while staying simple: easy to start with, easy to understand, and easy to operate on smaller screens. Inspired by the Much. Matter. In. Motion. (MMM) modeling space [54], which supports the modeling of many physics or chemical phenomena in a single block-based space, our domain-plural programming space supports multiple possible programming goals while refraining from exposing too many blocks or technical details to learners. We also focus more on the programming part: learners are not asked to exactly replicate a reference phenomenon. Rather, they are encouraged to use ABP for expressing themselves.

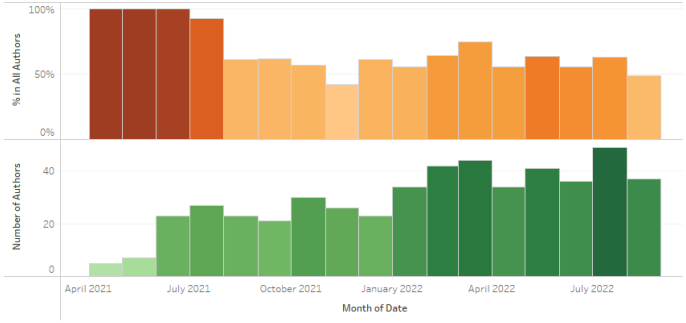

Figure 4: A subset of programming blocks in the Pocketworld Playground.

Fig. 4 presents a subset of programming blocks available in the POP. To determine the blocks to start with, we re-implemented 5 classical agent-based programs from NetLogo's models library. The Sunflower [70], Slime [67], and Paths [30] programs were chosen due to 1) their popularity among TU's learners; 2) they are representative of what ABP could achieve in generative arts, science, and social science; and 3) they are relatively easy to program in an agent-based way. The Circle and Random Walk programs were chosen from a widely-used ABM textbook [63] to introduce the basic ideas of ABP. While most blocks correspond with a single NetLogo command, a number of them were designed as micro-behaviors [37]: for example, "Find a Goal" block is designed so that the agent could store a random location. Then, the agent could pick up the information by using the "Turn towards a Goal" block. We also added blocks that would complete the expression space. For example, the "Teleport" block was introduced to complete the "Teleport Randomly" block. Finally, during the open beta, we accepted several proposals from learners that would increase the expression power. For instance, "With Shape" is introduced to allow conditional operation on agents with a given shape. For those micro-behavior blocks, the corresponding NetLogo code is available through TU's built-in code editor, and it is also possible for learners to change their implementation.

Overall, to promote a decentralized, probabilistic mindset rather than a deterministic one [66], our block design ensures that: 1) there should be no direct approach to operate on any single agent; and 2) randomness should be implicitly and explicitly represented across blocks. Randomness is explicitly designed when the "Randomly" suffix is present: "Turn Randomly", "Forward Randomly", etc. It could also be implicit: following NetLogo's own design decision, "Create Turtles" by default spawns a group of movable agents with random colors and directions.

3.3 Learner-Adaptable & Adaptive Scaffolds

While our block design only focused on the most necessary blocks, around 40 blocks remained in the design. Our second design goal is to present these blocks, together with the basic concepts of ABP, to learners that have no prior knowledge of ABP or programming at all. Meanwhile, some learners may have experience with Python or Scratch and might not be interested in another introductory tutorial. The wide and unknown levels of pre-implementation knowledge, as manifested in TU's learner discussion group, motivated us to make the second design decision:

> **Design Decision 2.** Our interactive scaffolds should recognize learners' agency in engagement, and support learners whenever they need or might be in need.

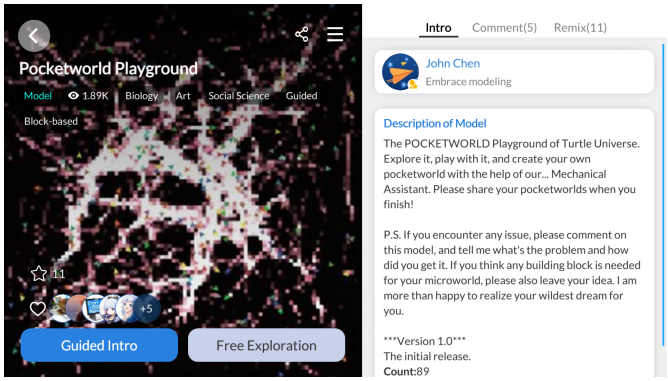

Figure 5: The interface before entering the POP.

The POP's interactive scaffolds are roughly designed in two parts: 1) Based on the 5 sample programs, we designed 2 guided pathways that introduce the foundational ideas of ABP; and 3 guided pathways that aim to help learners to design their first ABP project (see the next section); 2) Responsive scaffolds. For example, when a learner clicks on the "help" icon of each block, a short prompt will be displayed (learner-adaptable); when a learner spawns too many turtles, another prompt will be shown to guide the learner to revise the program (learner-adaptive).

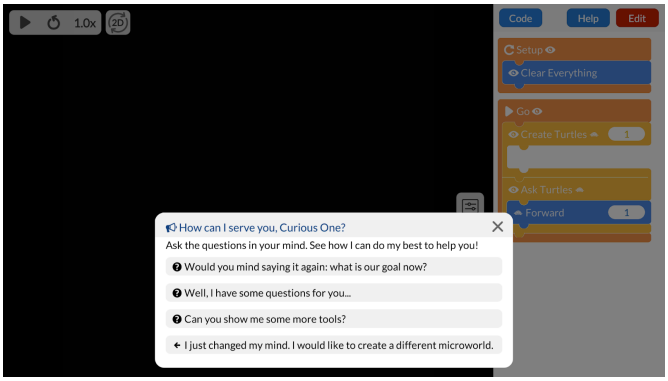

Figure 6: A possible version of the "Help" screen.

Figure 7: The "Intro" and "Create Your Own" pathways.

Through an integrated learner-adaptable and learner-adaptive design, we further develop [35]s conceptualization. In the POP, learners decide whether, how, and when to use scaffolds (or not). When entering the POP (Fig. 5), learners can choose from "Guided Intro", where both parts of the scaffolds will be activated; or "Free Exploration", where no guiding prompts will be shown. In either case, learners have the option to ask for additional support through the "Help" screen (Fig. 6), which adapts itself based on where learners are in (or out of) guided pathways. Here, the "Help" screen is both adaptive and adaptable: learners could either find new "tools" (blocks) for their projects, ask for hints, or switch to another pathway. Like all other prompts, learners could hide the screen at any moment without interrupting their current goals.

3.4 Personal and Communal Relevance

A key difference between Turtle Universe and previous ABM & P learning spaces is that it integrates more tightly with an online learning community. Different from the Modeling Commons [40], which is designated as a computational modeling community, TU focuses on a younger audience with more diverse interests. Learners who come to TU might have no idea what computational or agent-based modeling is. The different situation, therefore, asks for a different design decision.

> **Design Decision 3.** Multiple guided pathways should be personally relatable to individual learners and the learning community as a whole, while representative of what ABP could achieve.

The 5 classical agent-based programs, chosen as examples for our immersive ABP environment (see the previous section), served as the foundation of our guided pathways. The first divergence happens when we probe learners' prior knowledge (Fig. 7): if learners appear to have no prior knowledge (options 1, 2), they could choose the Circle and Random Walk pathways; otherwise, they could choose from the 3 pathways designed to help them build their own projects (Fig. 8). We intentionally chose generative arts (with Sunflower as the example), scientific modeling (with Slime), and social science modeling (with Path) to cover the most common applications of ABP. Games were excluded from the initial design to reduce the complexity of additional input blocks, but may come into the next design iteration as learners have started to request them.

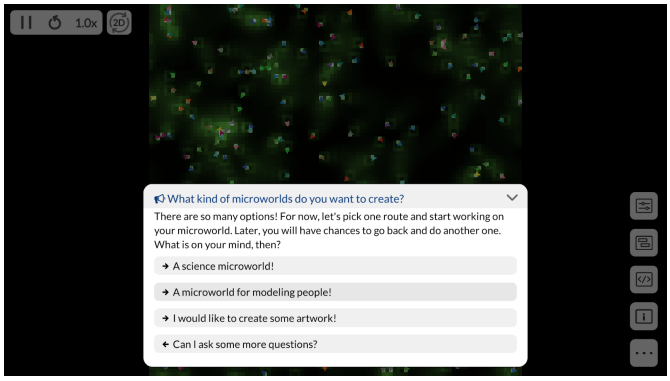

Figure 8: Three "Create Your Own" pathways.

Finally, a "Freestyle" option was added to each pathway: if learners are uninterested in the example, they have an explicit option to start their own project with less support.

Through the tone of our prompts, we seek to create the impression: that scaffolds are intended to assist learners in working towards their own goals and are subordinate to their will. Our design encourages learners to share their projects once they finished any "Create Own Project" pathway. The list of shared remixes (Fig. 5) further set up examples for learners to help develop the Turtle Universe's Online Community (TUOC).

4 EMPIRICAL STUDY

In 2021-2022, we conducted an empirical study of the Pocketworld Playground (POP) in a 14-month period. We collected and analyzed two datasets: 1) A log dataset from learners who explicitly consented to anonymized log data collection and could opt out at any point. As we only focus on learners who started engaging TU after the data collection began and used POP at least once, 14,442 learners' 30,390 sessions were included in our dataset. We did not collect personally identifiable data from learners. The timing of user interaction suggests that most learners were K-12 age learners in out-of-school contexts: most of them used Turtle Universe in school holidays, lunch breaks, or early nights. 2) A publicly shared artifact dataset from Turtle Universe's Online Community (TUOC)

that comprises all learners' shared projects created by the POP. By comparing each project with its remixing parent project, we used a previously reported cutoff [17] to only include projects with sufficient changes (20%), as the POP contains pieces of supporting code that typically stays unchanged. As a result, 744 shared projects were included.

We analyzed our log dataset quantitatively. The design of TU and the POP (Fig 5) creates two natural (or quasi) experimental conditions: Opted Out (quasi-control group), with learners who opted out of guided pathways during their visit; and Engaged (quasi-experimental group), learners who engaged with guided pathways during their first visit to the POP. The Opted Out condition includes two groups: 1) learners who chose "Free Exploration" and thus disable the guided pathways, 6,363 learners (44.1%); 2) learners who chose "Guided Intro" but did not go deep enough to engage with the guided pathways, 1,658 learners (11.5%). The Engaged condition, with learners who chose "Guided Intro" and engaged with the guided pathways, includes 6,421 learners (44.5%).

To measure the impact of our guided pathways on learners' situational and individual interests (RQ2 & 3), we used behavioral measures suggested by [46], namely meaningful engagement for measuring situational, short-term interest, and voluntary re-engagement for individual, long-term interest. Specifically, we used the following measures on learners' first engagement with the POP to understand our design's short-term impact on engagement and situational interest:

- **First-Time Time of Engagement**, to measure learners' short-term engagement with the activity;
- **First-Time Time of Engagement, Excluding Prompts**, to understand if learners' engagement increased with the POP, instead of only reading the prompts;
- **Meaningful Engagement Events**, calculated by the number of unique blocks learners added, plus if the learner re-ordered blocks (+1), removed blocks (+1), or shared with the community (+1).

To understand the impact of the "Help" screen, we went through each learner's logs. For each condition, we measure the effect on the time of engagement by separating learners who engaged with the screen from learners who did not. To measure learners' individual interests with the POP, we used their numbers of voluntary re-engagement. For learners who re-engaged at least once, we also measured their engagement time after the first engagement. Based on the distribution of re-engagement, we used 5 as the threshold for frequent learners (6.7% of the population).

Finally, we leveraged the shared artifact dataset. To understand if learners engaged in ABP practices or not (RQ1), we used a previously reported methodology to programmatically measure open-ended ABP practices learners' shared projects [17]. To understand the goals and patterns of learners' projects (RQ1), we qualitatively coded 60 randomly sampled projects (8.1%) and manually verified the automatic method with those projects. 15 projects (25%) were excluded since they are intended for non-project purposes: in TUOC, a learner may share a project to make an announcement, write about a piece of scientific knowledge, or start a discussion. Table 1 presents parts of our codebook used in this study. In addition, we coded projects to understand if they are graphically symmetric or clearly derived from one of the five guided pathway examples.

Table 1: Parts of our qualitative codebook used in this study

Category	**The category of the project.**
Drawing	The project is created as a (mostly) static drawing.
Animation	The project is created as an animation.
Proto-Model	The project has characteristics of an agent-based model.
Random	**How the project deals with randomness.** (Except for the randomness of colors)
Deterministic	The project has a (mostly) deterministic sequencing.
Implicit	The randomness of the project is implicit (e.g. in create-turtles or in the encapsulated functions like 'find-a-goal').
Explicit	The randomness of the project is explicit (e.g. the usage of 'random-float').

5 FINDINGS

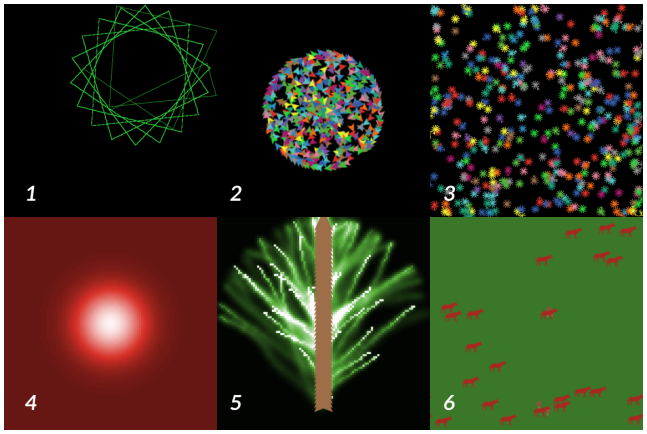

Figure 9: Screenshot of 6 projects made by learners. 1: "A Strange Shape". 2: "Real 3D Ball". 3: "Snowflake Everywhere". 4: "Explosion 1.0" 5: "Tree Simulation" 6: "Sheep Eat Wolves".

5.1 An Immersive ABP Environment

Did our learning design support learners' engagement with the ABP paradigm? (**RQ1**) We found learners engaged with decentralized and probabilistic programming and leveraged ABP for diverse personal goals.

Learners' shared projects revealed their widespread adoption of ABP paradigms. Our automatic analysis shows that most of the projects involve different levels of decentralized programming. 44% operate on multiple agents with the same computational rules; 34% also apply computational rules conditionally; 14% involve direct communication between agents. Only 8% come without decentralized practices: most of them operate on a single agent at a

Table 2: Effectiveness of Guided Pathways on Learners' First Engagement

	Duration (Sec)	**Excluding Prompts (Sec)**	**Meaningful Events**
Engaged	222.4	120.5	2.44
Opted Out	87.2	78.4	1.12
(Prob > F)	0.000	0.000	0.000

Table 3: The Fading Effect of Guided Pathways on Learners' Meaningful Events

Meaningful Events	**1st**	**2nd**	**3rd**	**4th**	**5th**
Engaged	2.44	3.26	3.94	3.96	4.17
Opted Out	1.12	2.19	2.91	3.44	3.51
(Cohen's d)	0.45	0.27	0.23	0.11	0.10
(Prob > F)	0.000	0.000	0.000	0.035	0.115
(% Engaged)	44.46%	45.36%	47.49%	44.76%	43.95%

Table 4: The Impact of the "Help" Screen on Learners' First-Time Engagement with the POP

Engagement (Seconds)	**Engaged**	**Opted Out**
Used "Help"	422.7	215.9
Not Used "Help"	149.3	78.4
(Prob > F)	0.000	0.000
(% Used)	23.1	9.4
(Time after "Help")	227.7	119

Table 5: Effectiveness of Guided Pathways on Learners' Voluntary Re-engagement

% Learners	**Re-engaged**	**Frequently Re-engaged**
Engaged	40.2	8.5
Opted Out	30	5.2
(Prob > Chi^2)	0.000	0.000

time or are essentially empty. 93% of authors shared at least one project with decentralized programming, while 20% of them shared at least one without.

Our qualitative coding supports this trend. Only 3 out of 45 projects (6.7%) did not engage in decentralized practices (e.g. Fig. 9-1). 5 out of 45 projects (11.1%) did not engage with randomness: except for one project, they were based on the Sunflower example (e.g. Fig. 9-2), which does not involve randomness as well. Most projects (26, 57.8%) explicitly conduct probabilistic programming, while most of the rest implicitly leverage it (14, 31.1%). Learners were surprisingly capable of leveraging randomness to create diverse types of symmetry. We identified 60% (24) of projects with randomness to be circular, square, horizontal, or diagonal symmetrical.

Learners leveraged ABP for diverse personal goals. We found that most projects were created as Animation (32, 77.1%), followed by Proto-Models (11, 24.4%) and Drawings (2, 4.4%). Different from the impression created by Fig. 9, most shared projects are actually animated. Leveraging emergence, Animation projects (e.g. Fig. 9-2,3,4) often create dazzling visual effects that are difficult to reproduce with non-ABP programming paradigms. Proto-Models (e.g. Fig. 9-5,6) are also animations, but with different stated purposes given by learners through project titles and descriptions. While they are often not scientific "models" in a strict sense, we found learners utilize computational rules beyond purely aesthetic reasons. Some learners even wrote descriptions that mimic the format of the "Info Tab" of ABMs in NetLogo's models library. Static drawings (e.g. Fig. 9-1) are rare and similar to Logo Geometry. Finally, while a few projects (9, 20%) are likely direct remixes of the five examples, most authors give their projects a new meaning (e.g. Fig. 9, a likely remix of Random Walk).

5.2 Learner-Adaptable & Adaptive Scaffolds

Did our learning design support learners' situational (or short-term) interests in ABP? (**RQ2**) We found both our guided pathways and the "Help" screen supported learners' situational interests in ABP.

The guided pathways effectively supported learners' situational interests in the POP during their first engagement. Table 2 shows the result of a one-way ANOVA analysis of our log dataset. Compared with learners who opted out, learners who engaged with the guided pathways have significant increases in the time of engagement, time of engagement with prompts excluded, and the number of meaningful engagement events.

The effects of our guided pathways gradually faded as learners engaged more with the POP. Additional ANOVA analyses indicate that in later engagement, while the number of meaningful events increased in both conditions, the effect sizes of the guided pathways gradually diminished (Table 3. The proportions of learners who engaged with guided pathways remained relatively unchanged. A possible explanation is that a higher percentage of learners who opted out of the guided pathways during the first engagement engaged with them later (54% among learners who engaged again) compared to those in the opposite situation (35%).

The learner-adaptable and adaptive "Help" screen significantly helped learners' short-term engagement with the POP. Our ANOVA analysis (4) shows similar significant results for both conditions. A more nuanced analysis shows that the engagement gains mostly come after the learners' usage of the "Help" screen. However, particularly in the "Opted Out" condition, only a limited portion of learners found and used the "Help" screen.

5.3 Personal and Communal Relevance

Did our learning design support learners' individual and collective interests in ABP? (**RQ3**) We found our guided pathways supported first-time learners' individual interests in the learning environment and supported the online ABP community's development.

Table 6: Effectiveness of Guided Pathways on Re-engaged Learners' Future Engagement

	Duration (Sec)	Excluding Prompts (Sec)	Meaningful Events
First-time Engaged	824.7	668.2	13.83
First-time Opted Out	597.3	476.2	9.29
(Prob > F)	0.000	0.000	0.000

Our guided pathways supported first-time learners' individual interests in the POP. (Table 5 A logistic regression analysis shows that engagement with guided pathways significantly increased the likelihood of voluntary (frequent) re-engagement with the POP in the 14-month period. Even within the Opted Out condition, learners who first chose "Guided Intro" and thus saw the guided pathways were slightly more likely to re-engage (32.4%; 6.0% frequently), compared to learners who chose "Free Exploration" and did not see them (29.4%; 4.7% frequently). Additionally, among learners who re-engaged with the POP, we saw first-time engagement with guided pathways led to improved future engagement metrics (Table 6).

The Pocketworld Playground played a crucial and unique role in bootstrapping TU's Online Community (TUOC). Throughout the 14-month period after the POP's release, most original shared projects in TU were created using the POP, and most authors shared original projects created using the POP (Fig. 10). Overall, 75% (207) of authors who shared original projects in TU shared at least once using the POP. Of the 135 authors who shared more than one POP project in the community, 43.0% (58) shared original text-based NetLogo projects. The log dataset further shows that most authors (81.5%) engaged with guided pathways at least once and most (91.4%) shared projects after engaging with guided pathways.

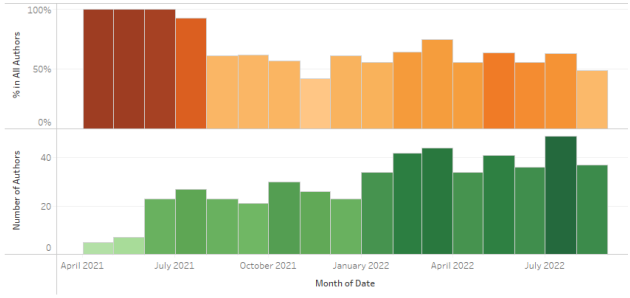

Figure 10: Authors of Original Projects in TU (Green); % of Original Authors Shared with the POP (Orange)

6 DISCUSSION

In this study, we presented the novel design of the Pocketworld Playground (POP) that aims to engage online, out-of-school learners with Agent-based Programming (ABP) in Turtle Universe (TU), the mobile version of NetLogo. Our answers to the three RQs that aim to evaluate the effectiveness of the design were positive. While our datasets, primarily focused on learners' behaviors and outputs, are not sufficient to provide an account of how the learning happened, in the following section, we discuss three design lessons that could help both our future design, and other learning designers that are interested in designing for learning ABM & P; or designing in online, out-of-school learning contexts.

Our first lesson was learned from the Proto-Models created by learners. The difference between what we coded as Proto-Model and a (strictly speaking) agent-based model (ABM) could be summarized as 1) an ABM usually accepts different parameters as inputs explicitly, and 2) an ABM often leverages graphs to visualize the macro-level patterns of the model. However, both were almost impossible to implement with the block-based environment of the POP. It was not that learners were not willing or not advanced enough to make ABMs; there was simply no support for them. In other words, we underestimated how far learners advanced on their own. Therefore, when designing constructionist learning environments, it is important for future designs to recognize learners' potential and be responsive to their emergent needs in the long run.

Our second lesson was related to the "Help" screen. While it was successful in supporting learners who engaged with the screen, not many learners found it: less than a quarter in the "Engaged" condition, and less than 1/10 in the "Opted Out" condition. One reason could be the screen hides deep at the corner of the interface. Unless the learner intentionally clicks on the "Help" button, the screen would not pop out. Considering that the "Help" screen sustained so many essential design features of the learning environment - where learners could switch to another pathway, and where learners could ask for more blocks - a future design would need to raise learners' awareness of the feature and more proactively suggest the usage of it, whenever learners are identified as possibly stuck.

Our final lesson came as a surprise. While we assumed that first-time learners would more likely choose "Guided Intro" over "Free Exploration", in reality, the portion of learners who chose "Guided Intro" peaked at the third visit and only started to decline thereafter. Our current understanding is that in online, out-of-school learning contexts, first-time learners could be more interested in self-driven exploration. Unfortunately, when they realized that more support is needed, our design did not provide a pathway from "Free Exploration" to guided pathways. While they could exit the environment and choose again, a better design should recognize their needs and encourage them to use scaffolds when needed. It is also possible that the names we chose for the options tilted toward "Free Exploration" and need revision.

Finally, we recognize the limitations of our research methodology which was partially the result of a world ravaged by COVID and school shutdown. An extended study would need to focus more on individual trajectories of learners' engagement with the learning environment through interviews, observations, and also analyzing log datasets of learners. It also needs to evaluate the proposed design changes based on our findings. While log data analysis generated much statistical power for this study, interviews and observations could generate much more meaningful hypotheses about how the design worked beyond our initial design ideas.

7 SELECTION AND PARTICIPATION OF CHILDREN

We informally co-designed with online youth (approx. 13-18 year-olds) in Turtle Universe, an app that is publicly available in mobile app stores. We informally discussed design ideas and decisions through public online chat groups (children as informants). Youth also posted feedback or feature requests on the first author's message board (children as designers). Their ideas are iteratively integrated into the learning design, and the data collection starts after the version is finalized. Youth voluntarily participate in our learning activity through the online software, and there was no selection process. Following a protocol approved by the authors' University IRB, we anonymously collected log data from all assented users in Turtle Universe, as well as shared projects that are publicly available on the app.

REFERENCES

[1] Golnaz Arastoopour Irgens, Sugat Dabholkar, Connor Bain, Philip Woods, Kevin Hall, Hillary Swanson, Michael Horn, and Uri Wilensky. 2020. Modeling and Measuring High School Students' Computational Thinking Practices in Science. *Journal of Science Education and Technology* 29, 1 (Feb. 2020), 137–161. https://doi.org/10.1007/s10956-020-09811-1

[2] MIT Education Arcade. 2023. Starlogo TNG Curriculum. https://education.mit.edu/project/starlogo-tng/

[3] Sara Atske. 2022. Teens, Social Media and Technology 2022. https://www.pewresearch.org/internet/2022/08/10/teens-social-media-and-technology-2022/

[4] Steven C. Bankes. 2002. Agent-based modeling: A revolution? *Proceedings of the National Academy of Sciences* 99, suppl_3 (May 2002), 7199–7200. https://doi.org/10.1073/pnas.072081299 Publisher: Proceedings of the National Academy of Sciences.

[5] Satabdi Basu, Pratim Sengupta, and Gautam Biswas. 2015. A Scaffolding Framework to Support Learning of Emergent Phenomena Using Multi-Agent-Based Simulation Environments. *Research in Science Education* 45, 2 (April 2015), 293–324. https://doi.org/10.1007/s11165-014-9424-z

[6] Amanda Bell. 2020. Two Approaches to Teaching with NetLogo: Examining the Role of Structure and Agency. In *The Interdisciplinarity of the Learning Sciences, 14th International Conference of the Learning Sciences (ICLS) 2020*, Vol. 4.

[7] Marina Umaschi Bers. 2012. *Designing digital experiences for positive youth development: From playpen to playground.* OUP USA.

[8] Corey Brady. 2021. Patches as an expressive medium for exploratory multi-agent modelling. *British Journal of Educational Technology* 52, 3 (2021), 1024–1042. Publisher: Wiley Online Library.

[9] Corey Brady, Melissa Gresalfi, Selena Steinberg, and Madison Knowe. 2020. Debugging for Art's Sake: Beginning Programmers' Debugging Activity in an Expressive Coding Context. (June 2020). https://repository.isls.org//handle/1/6319 Publisher: International Society of the Learning Sciences (ISLS).

[10] Corey Brady, Kai Orton, David Weintrop, Gabriella Anton, Sebastian Rodriguez, and Uri Wilensky. 2017. All Roads Lead to Computing: Making, Participatory Simulations, and Social Computing as Pathways to Computer Science. *IEEE Transactions on Education* 60, 1 (Feb. 2017), 59–66. https://doi.org/10.1109/TE.2016.2622680 Conference Name: IEEE Transactions on Education.

[11] Karen Brennan and Mitchel Resnick. 2013. Stories from the scratch community: Connecting with ideas, interests, and people. In *Proceeding of the 44th ACM technical symposium on Computer science education*. 463–464.

[12] Karen A. (Karen Ann) Brennan. 2013. *Best of both worlds : issues of structure and agency in computational creation, in and out of school.* Thesis. Massachusetts Institute of Technology. https://dspace.mit.edu/handle/1721.1/79157 Accepted: 2013-06-17T19:03:09Z Journal Abbreviation: Issues of structure and agency in computational creation, in and out of school.

[13] Amy Susan Bruckman. 1997. *MOOSE crossing: Construction, community and learning in a networked virtual world for kids.* PhD Thesis. Massachusetts Institute of Technology.

[14] Rafael C. Cardoso and Angelo Ferrando. 2021. A Review of Agent-Based Programming for Multi-Agent Systems. *Computers* 10, 2 (Feb. 2021), 16. https://doi.org/10.3390/computers10020016 Number: 2 Publisher: Multidisciplinary Digital Publishing Institute.

[15] John Chen and Uri Wilensky. 2020. NetLogo Mobile: An Agent-Based Modeling Platform and Community for Learners, Teachers, and Researchers. In *Proceedings of International Conference of the Learning Sciences 2020.*

[16] John Chen and Uri J. Wilensky. 2021. Turtle Universe. https://turtlesim.com/products/turtle-universe/

[17] John Chen and Uri J. Wilensky. 2023. Measuring Young Learners' Open-ended Agent-based Programming Practices with Learning Analytics. In *Proceedings of AERA Annual Meeting 2023.*

[18] John Chen and Uri J. Wilensky. 2023. Tortuga: Building Interactive Scaffolds for Agent-based Modeling and Programming in NetLogo. In *Proceedings of ISLS Annual Meeting 2023.*

[19] John Chen and Lexie Zhao. 2017. Physics Lab AR. https://turtlesim.com/products/physics-lab/

[20] Kar-Hai Chu, Rachel Goldman, and Elizabeth Sklar. 2005. Roboxap: an agent-based educational robotics simulator. In *Agent-based Systems for Human Learning Workshop at AAMAS-2005.*

[21] Allan Collins, John Seely Brown, and Ann Holum. 1991. Cognitive apprenticeship: Making thinking visible. *American educator* 15, 3 (1991), 6–11.

[22] Rodrigo da Silva Guerra, Joschka Boedecker, Hiroshi Ishiguro, and Minoru Asada. 2007. Successful Teaching of Agent-Based Programming to Novice Undergrads in a Robotic Soccer Crash Course. (2007), 21. Publisher: Citeseer.

[23] Amanda Dickes, Amy Farris, and Pratim Sengupta. 2016. Integrating agent-based programming with elementary science: The role of sociomathematical norms. *arXiv preprint arXiv:1609.00850* (2016).

[24] Joshua M. Epstein and Robert Axtell. 1996. *Growing artificial societies: social science from the bottom up.* Brookings Institution Press.

[25] Amy Voss Farris and Pratim Sengupta. 2014. Perspectival computational thinking for learning physics: A case study of collaborative agent-based modeling. *arXiv preprint arXiv:1403.3790* (2014).

[26] Ferran Ferrer, Esther Belvís, and Jordi Pàmies. 2011. Tablet PCs, academic results and educational inequalities. *Computers & Education* 56, 1 (Jan. 2011), 280–288. https://doi.org/10.1016/j.compedu.2010.07.018

[27] Deborah A. Fields, Michael Giang, and Yasmin Kafai. 2014. Programming in the wild: trends in youth computational participation in the online scratch community. In *Proceedings of the 9th workshop in primary and secondary computing education*. 2–11.

[28] Aristotelis Gkiolmas, Kostas Karamanos, Anthimos Chalkidis, Constantine Skordoulis, Maria Papaconstantinou, and Dimitrios Stavrou. 2013. Using Simulations of NetLogo as a Tool for Introducing Greek High-School Students to Eco-Systemic Thinking. *Advances in Systems Science and Applications* 13, 3 (Sept. 2013), 276–298. https://ijassa.ipu.ru/index.php/ijassa/article/view/141 Number: 3.

[29] Gary Greenfield and Penousal Machado. 2015. Ant-and ant-colony-inspired alife visual art. *Artificial life* 21, 3 (2015), 293–306. Publisher: MIT Press.

[30] R. Grider and Uri J. Wilensky. 2015. NetLogo Paths model. Center for Connected Learning and Computer Based Modeling, Northwestern University. https://ccl.northwestern.edu/netlogo/models/Paths

[31] Volker Grimm, Eloy Revilla, Uta Berger, Florian Jeltsch, Wolf M. Mooij, Steven F. Railsback, Hans-Hermann Thulke, Jacob Weiner, Thorsten Wiegand, and Donald L. DeAngelis. 2005. Pattern-oriented modeling of agent-based complex systems: lessons from ecology. *science* 310, 5750 (2005), 987–991. Publisher: American Association for the Advancement of Science.

[32] Cindy E. Hmelo-Silver, Lei Liu, Steven Gray, and Rebecca Jordan. 2015. Using representational tools to learn about complex systems: A tale of two classrooms. *Journal of Research in Science Teaching* 52, 1 (2015), 6–35. https://doi.org/10.1002/tea.21187 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/tea.21187.

[33] Michael S. Horn, Jeremy Baker, and Uri J. Wilensky. 2020. NetTango Web. https://netlogoweb.org/nettango-builder

[34] Michael S. Horn, David Weintrop, and Emily Routman. 2014. Programming in the pond: A tabletop computer programming exhibit. In *CHI'14 Extended Abstracts on Human Factors in Computing Systems*. 1417–1422.

[35] Shari L. Jackson, Joseph Krajcik, and Elliot Soloway. 1998. The design of guided learner-adaptable scaffolding in interactive learning environments. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 187–194.

[36] Yasmin B. Kafai. 2016. From computational thinking to computational participation in K–12 education. *Commun. ACM* 59, 8 (July 2016), 26–27. https://doi.org/10.1145/2955114

[37] Ken Kahn and Howard Noble. 2010. The modelling4all project a web-based modelling tool embedded in web 2.0. In *2nd International ICST Conference on Simulation Tools and Techniques.*

[38] A. Kashihara, M. Shinya, K. Taira, and K. Sawazaki. 2008. Cognitive Apprenticeship Approach to Developing Meta-Cognitive Skill with Cognitive Tool for Web-based Navigational Learning. ACTA Press. https://www.actapress.com/Abstract.aspx?paperId=32882

[39] Irene Lee, Fred Martin, Jill Denner, Bob Coulter, Walter Allan, Jeri Erickson, Joyce Malyn-Smith, and Linda Werner. 2011. Computational thinking for youth in practice. *Acm Inroads* 2, 1 (2011), 32–37. Publisher: ACM New York, NY, USA.

[40] R. Lerner, Sharona T. Levy, and Uri Wilensky. 2010. Connected modeling: design and analysis of the modeling commons. In *International Journal of E-Learning and Learning Objects (submitted to IJELLO special series of Chais Conference 2010 best papers).*

[41] Sharona T. Levy and Uri Wilensky. 2005. An analysis of student patterns of exploration with NetLogo models embedded in the connected chemistry environment. In *Proceedings of the annual meeting of the American Educational Research Association*.

[42] Thomas March. 2007. Revisiting WebQuests in a Web 2 World. How developments in technology and pedagogy combine to scaffold personal learning. *Interactive educational multimedia: IEM* (2007), 1–17. https://raco.cat/index.php/IEM/article/view/205331

[43] Kit Martin, Michael Horn, and Uri Wilensky. 2018. *Ant Adaptation: A Complex Interactive Multitouch Game about Ants Designed for Museums*.

[44] Elliott A. Medrich. 1982. *The Serious Business of Growing Up: A Study of Children's Lives Outside School*. University of California Press.

[45] Shari Metcalf, Amanda Dickes, Karen Brennan, and Chris Dede. 2019. Design of an Agent-Based Visual Programming Tool for Elementary Ecosystem Science Learning. In *Proceedings of the 2019 ACM Conference on International Computing Education Research (ICER '19)*. Association for Computing Machinery, New York, NY, USA, 309. https://doi.org/10.1145/3291279.3341210

[46] Joseph E. Michaelis and David Weintrop. 2022. Interest Development Theory in Computing Education: A Framework and Toolkit for Researchers and Designers. *ACM Transactions on Computing Education (TOCE)* (2022). Publisher: ACM New York, NY.

[47] Seymour Papert. 1980. Mindstorms: Children, computers, and powerful ideas. (1980). Publisher: Basic Books.

[48] Mark Phillips, Jackie Scolaro, and Daniel Scolaro. 2010. The Emergence of Agent-Based Technology as an Architectural Component of Serious Games. https://ntrs.nasa.gov/citations/20100012860 NTRS Author Affiliations: VP Business Development NTRS Document ID: 20100012860 NTRS Research Center: Langley Research Center (LaRC).

[49] Mark Pogson, Rod Smallwood, Eva Qwarnstrom, and Mike Holcombe. 2006. Formal agent-based modelling of intracellular chemical interactions. *Biosystems* 85, 1 (July 2006), 37–45. https://doi.org/10.1016/j.biosystems.2006.02.004

[50] K. Ann Renninger and Suzanne E. Hidi. 2015. *The power of interest for motivation and engagement*. Routledge.

[51] Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, and Yasmin Kafai. 2009. Scratch: programming for all. *Commun. ACM* 52, 11 (Nov. 2009), 60–67. https://doi.org/10.1145/1592761.1592779

[52] Gabriela T. Richard and Yasmin B. Kafai. 2016. Blind Spots in Youth DIY Programming: Examining Diversity in Creators, Content, and Comments within the Scratch Online Community. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. Association for Computing Machinery, New York, NY, USA, 1473–1485. https://doi.org/10.1145/2858036.2858590

[53] Ricarose Roque, Natalie Rusk, and Mitchel Resnick. 2016. Supporting Diverse and Creative Collaboration in the Scratch Online Community. In *Mass Collaboration and Education*, Ulrike Cress, Johannes Moskaliuk, and Heisawn Jeong (Eds.). Springer International Publishing, Cham, 241–256. https://doi.org/10.1007/978-3-319-13536-6_12

[54] Janan Saba, Hagit Hel-Or, and Sharona T. Levy. 2020. " When is the pressure zero inside a container? Mission impossible" 7th grade students learn science by constructing computational models using the much. matter. in. motion platform. In *Proceedings of the Interaction Design and Children Conference*. 293–298.

[55] Pratim Sengupta, John S. Kinnebrew, Satabdi Basu, Gautam Biswas, and Douglas Clark. 2013. Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies* 18, 2 (June 2013), 351–380. https://doi.org/10.1007/s10639-012-9240-x

[56] Yoav Shoham. 1993. Agent-oriented programming. *Artificial intelligence* 60, 1 (1993), 51–92. Publisher: Elsevier.

[57] Mario Soflano, Thomas M. Connolly, and Thomas Hainey. 2015. An application of adaptive games-based learning based on learning style to teach SQL. *Computers & Education* 86 (Aug. 2015), 192–211. https://doi.org/10.1016/j.compedu.2015.03.015

[58] Xiaodan Tang, Yue Yin, Qiao Lin, Roxana Hadad, and Xiaoming Zhai. 2020. Assessing computational thinking: A systematic review of empirical studies. *Computers & Education* 148 (April 2020), 103798. https://doi.org/10.1016/j.compedu.2019.103798

[59] Seth Tisue and Uri Wilensky. 2004. Netlogo: A simple environment for modeling complexity. In *International conference on complex systems*, Vol. 21. Boston, MA, 16–21.

[60] Ioanna Vekiri. 2010. Socioeconomic differences in elementary students' ICT beliefs and out-of-school experiences. *Computers & Education* 54, 4 (May 2010), 941–950. https://doi.org/10.1016/j.compedu.2009.09.029

[61] David Weintrop, Elham Beheshti, Michael Horn, Kai Orton, Kemi Jona, Laura Trouille, and Uri Wilensky. 2016. Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology* 25, 1 (Feb. 2016), 127–147. https://doi.org/10.1007/s10956-015-9581-5

[62] U. Wilensky. 2020. Restructuration theory and agent-based modeling: Reformulating knowledge domains through computational representations. Pages: 287–300 Publication Title: Designing Constructionist Futures: The Art, Theory, and Practice of Learning Designs.

[63] Uri Wilensky and Seymour Papert. 2010. Restructurations: Reformulations of knowledge disciplines through new representational forms. *Proceedings of Constructionism* 17 (2010), 1–15.

[64] Uri Wilensky and William Rand. 2015. *An introduction to agent-based modeling: modeling natural, social, and engineered complex systems with NetLogo*. Mit Press.

[65] Uri Wilensky and Kenneth Reisman. 2006. Thinking like a wolf, a sheep, or a firefly: Learning biology through constructing and testing computational theories—an embodied modeling approach. *Cognition and instruction* 24, 2 (2006), 171–209.

[66] Uri Wilensky and Mitchel Resnick. 1999. Thinking in levels: A dynamic systems approach to making sense of the world. *Journal of Science Education and technology* 8, 1 (1999), 3–19. Publisher: Springer.

[67] Uri J. Wilensky. 1997. NetLogo Slime model. Center for Connected Learning and Computer Based Modeling, Northwestern University. https://ccl.northwestern.edu/netlogo/models/Slime

[68] Uri J. Wilensky. 1997. NetLogo Wolf Sheep Predation model. Center for Connected Learning and Computer Based Modeling, Northwestern University. http://ccl.northwestern.edu/netlogo/models/WolfSheepPredation

[69] Uri J. Wilensky. 1999. NetLogo. Center for Connected Learning and Computer Based Modeling, Northwestern University. http://ccl.northwestern.edu/netlogo/

[70] Uri J. Wilensky. 2003. NetLogo Sunflower model. Center for Connected Learning and Computer Based Modeling, Northwestern University. https://ccl.northwestern.edu/netlogo/models/Sunflower

[71] Michelle Hoda Wilkerson-Jerde and Uri Wilensky. 2010. Restructuring change, interpreting changes: The deltatick modeling and analysis toolkit. *Proceedings of constructionism* (2010).

[72] Jeannette M. Wing. 2006. Computational thinking. *Commun. ACM* 49, 3 (2006), 33–35. Publisher: ACM New York, NY, USA.

[73] S. Wu, Hillary Swanson, Bruce Sherin, and Uri J. Wilensky. 2023. Using Agent-based Computational Modeling Microworlds to Help Middle School Students Learn about Epidemiology. In *Proceedings of AERA Annual Meeting 2023*.

[74] Lin Xiang and Cynthia Passmore. 2015. A framework for model-based inquiry through agent-based programming. *Journal of Science Education and Technology* 24, 2 (2015), 311–329. Publisher: Springer.

[75] Franco Zambonelli and Andrea Omicini. 2004. Challenges and Research Directions in Agent-Oriented Software Engineering. *Autonomous Agents and Multi-Agent Systems* 9, 3 (Nov. 2004), 253–283. https://doi.org/10.1023/B:AGNT.0000038028.66672.1e