The Effect of Informing Agency in Self-Directed Online Learning Environments

Benjamin Xie
University of Washington
The Information School, DUB
bxie@uw.edu

Greg L. Nelson
University of Washington
The Allen School, DUB
glnelson@uw.edu

Harshitha Akkaraju
University of Washington
The Information School, DUB
akkarh@uw.edu

William Kwok
University of Washington
The Information School, DUB
wkwok16@uw.edu

Amy J. Ko
University of Washington
The Information School, DUB
ajko@uw.edu

ABSTRACT

Choices learners make when navigating a self-directed online learning tool can impact the effectiveness of the experience. But these tools often do not afford learners the agency or the information to make decisions beneficial to their learning. We evaluated the effect of varying levels of information and agency in a self-directed environment designed to teach programming. We investigated three design alternatives: informed high-agency, informed low-agency, and less informed high-agency. To investigate the effect of these alternatives on learning, we conducted a study with 79 novice programmers. Our results indicated that increased agency and information may have translated to more motivation, but not improved learning. Qualitative results suggest this was due to the burden that agency and information placed on decision-making. We interpret our results in relation to informing the design of self-directed online tools for learner agency.

Author Keywords

agency; educational technology; interaction design

CCS Concepts

•**Human-centered computing** → **Interaction design theory, concepts and paradigms;** *Empirical studies in HCI;* •**Social and professional topics** → Computing education;

INTRODUCTION: DESIGN SPACE FOR AGENCY

Agency, or the sense we are in control of our actions and their effects [45, 42], is important to learning. Agency can make students contributors to their learning experience rather than just products of them [4]. In classroom settings, teachers overwhelmingly believed that affording students agency improves motivation and learning outcomes, while also recognizing that limits to agency were necessary [23]. This recognition that both freedom to make choices as well as the scaffolding to limit these choices suggests that designing to promote effective learner agency is important, but nuanced.

The need to balance freedom and guidance is especially true in self-directed learning settings, such as online tutorials and educational games, where designers create the entire instructional experience and no human teacher is available to provide assistance. In these experiences, agency can be framed as a phenomenon involving both a learner and their learning environment, in which the actions that learners desire are among those they can actually take [67]. The goal of having learners exert agency is to have learners make informed choices to support their engagement [8, 56], motivation [17], and learning [62, 63, 56, 17]. Agency might manifest as a learner deciding that an exercise is too easy, choosing to jump ahead to a more difficult exercise, or realizing that they lack some understanding, and reviewing some prior instruction. These decisions emerge from a learner having a goal, taking an available action to support their goal, and then reflecting on the result of their action [44].

Elements of self-directed learning environments will always influence learner agency, but not always in ways that benefit learning. To exert agency, learners must first perceive that they can do so [43, 14]. Learners rely on their perceptions of the environment to develop their sense of agency [67], so the design of the learning environment is impactful to their agency. Designers exert *indirect control* [59] over learner actions. These elements of indirect control can inadvertently result in learners following similar paths for no reason beneficial to learning, therefore unnecessarily limiting their agency. This was the case in a computer-based math game, where learners were afforded the agency to play mini-games in any order but instead tended to follow a dotted line which visually connected mini-games in a somewhat arbitrary order [26, 48], resulting in no difference in learning outcomes between high- and low-agency variations of this game. Therefore, designers must effectively scaffold a self-directed learning experience to

L@S ’20, August 12–14, 2020, Virtual Event, USA.

ACM ISBN 978-1-4503-7951-9/20/08 ...$15.00.
http://dx.doi.org/10.1145/3386527.3405928

ensure learners exert agency by making informed choices that benefit their learning.

Prior work on agency in self-directed learning environments has primarily explored the effect of more or less agency on learning. For example, studies of the self-directed educational game Crystal Island [56] have found that limiting available actions in the virtual environment led to better learning gains when compared to a high-agency condition, but limiting options also led to an increased propensity for guessing [64, 58]. However, prior work has also found that too much agency can also be detrimental [2]. This was the case in Chen et al. 2019, which found that learners with more prior knowledge in a high-agency condition (where they could choose their own preparation tasks) exhibited similarly unproductive behavior such as guessing [11]. These findings suggest that designing for agency means finding a "sweet spot" that brings the benefits of choice, while preventing learners from being overwhelmed [62, 19, 14].

While prior work on self-directed learning has explored varying levels of agency [11, 39, 40, 55] and agency over different aspects of learning [15, 14, 13], it has not jointly explored varying levels of *information* to support agency. And information is key: there is a difference between giving a learner a choice about what to do next, and giving them carefully designed information about the risks and opportunities of those choices. Prior theoretical work calls this *proximal action-related information* [44], which aims to help learners determine their 1) capacity to act (e.g. empty check boxes indicating practice that has not been completed, showing available mini-games and hiding previously completed ones), 2) current ability to do so (e.g. skill bars showing estimated knowledge in an open learner model [28], earned badges to reflect accomplishments), and 3) the predicted result of taking an action (e.g. an adaptive recommendation denoting that a specific practice question can serve as review). This framework suggests how systems might provide such information, but prior work provides no design guidance on the effects of varying levels of action-related information on agency.

To contribute to this design guidance, we built a self-directed learning environment for learning Python programming, varying both the amount of agency afforded and the amount of information provided to support learning decisions. We specifically studied three design variations: 1) informed (high-information) high-agency, 2) uninformed (low-information) high-agency, and 3) informed low-agency. With these alternatives, we then conducted a between-subjects experiment to investigate the effects of these design choices on 1) learners' experiences, and 2) learning outcomes. Participants in the study engaged in self-directed learning for a week, then took a survey and post-test measuring learning gains. In the rest of this paper, we discuss the design alternatives and our study design in detail, then present our results and their implications on designing for agency.

THEORETICAL BACKGROUND ON AGENCY

Before discussing our design alternatives and study design, we discuss the theoretical views that inform both. In particular, while there are many definitions of *agency*, in this paper we frame it as occurring when a learner can take actions that align with their learning-related goals [67]. Within this framing, we position Bandura's notion of *self-efficacy* as the primary individual factor that influences both learning and the use of proximal action-related information found in a learning environment [4, 3]. From this view, learners must believe in their abilities to organize and execute a course of action as well as process information from the environment regarding potential actions to take and their implications.

While agency is dependent on self-efficacy, acting upon self-efficacy requires information from a learning environment. We specifically draw upon frameworks of *proximal action-related information*, which positions information that is situated near and related to a decision [44] as critical to agency. Examples of such information include skill bars indicating the current state of understanding, check boxes indicating what a learner has or has not completed, or adaptive recommendations suggesting a next topic to learn.

Finally, we also draw upon the *Preference Construction* (PC) model of decision-making to explain the importance of proximal action-related information to agency. This model is commonly used in explaining economic decision-making and frames preferences as a contextually developed construct [5, 34]. PC draws upon Herbert Simon's notion of *bounded rationality*, which states that the complexity of a decision task, limitations of cognitive resources and knowledge of people, and the tendency to reduce decision effort lead to a limited rationality [61]. This implies a trade-off between decision-making effort and the accuracy of the decision outcome [51] and that because PC is contextual, it is susceptible to different kinds of biases. Within the context of recommender systems, influences such as context effects, primary/recency effects, framing effects, and anchoring effects may bias how people make decisions [41]. PC states that humans do not have a clear preference in the very beginning, but rather develop preferences within the context of a decision process. Therefore, proximal information is critical for exerting agency.

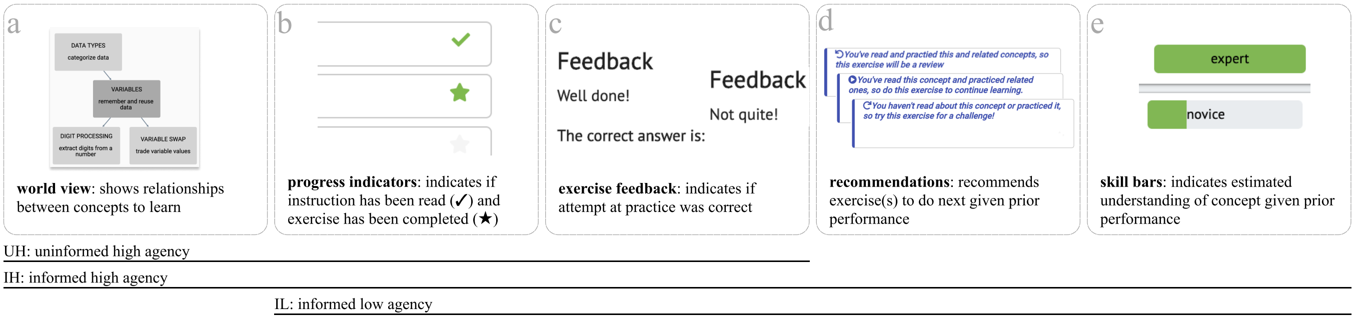
An aspect of agency that is beyond the scope of this paper is metacognition, one's ability to monitor and regulate their own cognitive processes, behavior, and affect [46]. Metacognitive skills can support agency [47, 44], but vary amongst novice programmers [38]. We attempted to remove this confound through random assignment in our study, detailed later in the paper.

THREE DESIGNS TO EXPLORE AGENCY

Given these theoretical foundations, we considered three variations on degrees on agency and proximal information: an Informed, High-Agency (*IH*) design that gave learners agency and information; an Informed, Low-Agency (*IL*) design that gave learners information but little choice; and Uninformed, High-Agency (*UH*), which gave less informed choices. In this section, we describe the learning domain, how we provided proximal information, and our three designs.

Learning Domain: Self-directed intro to Python

To explore agency and proximal information, we selected the domain of learning to program. As a domain, program-


a
DATA TYPES
categorize data
VARIABLES
remember and reuse data
DIGIT PROCESSING
extract digits from a number
VARIABLE SWAP
trade variable values
b
c
Feedback
Well done!
The correct answer is:
Feedback
Not quite!
d
You've read and practied this and related concepts, so this exercise will be a review
You've read this concept and practiced related ones, so do this exercise to continue learning.
You haven't read about this concept or practiced it, so try this exercise for a challenge!
e
expert
novice

world view: shows relationships between concepts to learn

progress indicators: indicates if instruction has been read (✓) and exercise has been completed (★)

exercise feedback: indicates if attempt at practice was correct

recommendations: recommends exercise(s) to do next given prior performance

skill bars: indicates estimated understanding of concept given prior performance

UH: uninformed high agency

IH: informed high agency

IL: informed low agency

Figure 1: Features of Codeitz designed to provide learners with proximal action-related information for deciding what to learn next. Variations of the environment exposed learners to different subsets of the features (see lines at bottom of figure).

ming has many attractive features: at this point in history, many people want to learn it; there are many examples of self-directed learning environments for learning to code online; and the domain itself has concepts with relatively clear inter-dependencies that are amenable to learner modeling.

To support our investigation, we developed *Codeitz*, a new self-directed learning environment to teach Python programming (see Fig. 1). All variations of Codeitz shared the same introductory curriculum adapted from the materials defined in [70]. This curriculum was designed to assume no prior programming knowledge and cover introductory Python concepts including basic constructs (data types, operators, variables, print statements, conditionals) and templates demonstrating ways to use what was learned (variable swap, float comparison, find min/max, digit processing).

While the original learning materials were created for more linear learning, we relaxed this constraint to make learning through exploration more feasible. Following semantic dependencies defined by the Python programming language and extending that pattern of hard dependencies to templates [70], we developed a concept hierarchy that learners could use to decide what to learn next (shown in Fig. 2 and described below). To adapt the learning materials to match the concept hierarchy we defined, we adjusted instructional content to assume learners only visited parent concepts and created additional exercises to practice which relied on fewer other concepts. We kept some examples and exercises which relied on concept dependencies not reflected in our hierarchy, so this adaptation was not complete.

From an instructional design perspective, we designed Codeitz to be a self-contained learning environment. To learn a concept, learners could read instruction to develop conceptual understanding of an aspect of Python and then attempt practice exercises where they received feedback related to correctness from the system. Practice exercises included multiple choice, short answer, filling in Memory Tables [71] to trace program state, and writing code. To support a formative experience, learners were able to retry practice exercises and see the answer whenever they wanted. Each page of instruction or exercise mapped to exactly one concept.

Three Codeitz Designs Varying Agency, Information

All three variations of Codeitz had the same instructional material and included conventional feedback on learning progress (Fig. 1b) and exercise correctness (Fig. 1c) common to online learning tools such as online courses (e.g. edX, Coursera) and learning platforms (e.g. Khan Academy, Codecademy). However, the designs varied in the amount of agency and predictive information afforded to learners.

We specifically focused on supporting learners' decision of *what to learn next* by varying the presence or absence of three features that either afforded agency or offered proximal information to support learning decisions. One feature was a **world view** showing Python concepts and their dependencies (Figures 1a, 2). We designed the world view to be as nonlinear as possible so as to encourage learners to exert agency and explore different concepts while having an understanding of their underlying relationships. Learners could use the world view to explore concepts as they relate to other concepts they may have already learned. Another feature was **recommendations** of what to learn next (Fig. 1d). These were based on the estimated difficulty of the exercise relative to learners' current levels of understanding for a concept. Recommendations supported the goals of *reviewing* (exercise involves a concept learner is knowledgeable with), *continuing* (exercise involves concept learner has made progress with), or *challenging* (exercise involves a concept a learner has little experience with). Learners can use recommendations to judge how certain exercises may support current goals. And finally, **skill bars** provided estimated levels of mastery for a concept (Fig. 1e), to help learners determine if they needed to complete all of the instruction and exercises or whether they could move on to another concept. Learners can use skill bars to judge how well they have mastered a specific skill. Our three designs offered unique combinations of these three features.

UH: The uninformed high-agency version lacked recommendations & skill bars, but still required learners to exert agency. We intended for this version to reflect an open online course (e.g. a MOOC) in the information provided to a learner as well as its availability of content. In this design, learners were uninformed of system predictions from their prior responses. They used information about the knowledge domain, progress they made, and exercise feedback (Fig. 1a-c) to guide their

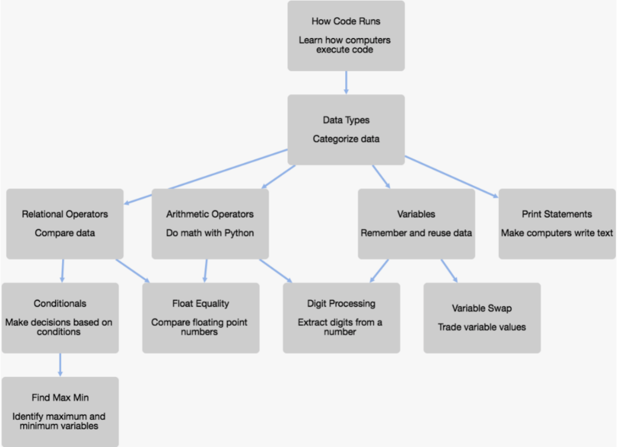
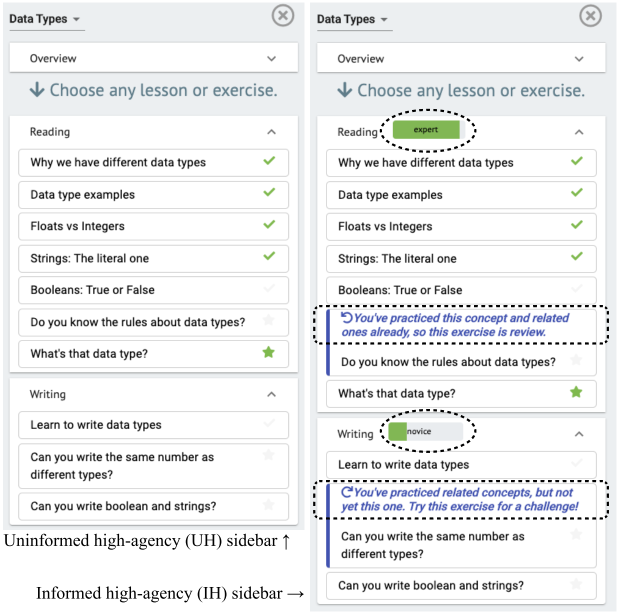


Figure 2: The world view, showing Python concepts taught and major dependencies between them.


Data Types ▾

Overview

↓ Choose any lesson or exercise.

Reading

Why we have different data types ✓

Data type examples ✓

Floats vs Integers ✓

Strings: The literal one ✓

Booleans: True or False

Do you know the rules about data types?

What's that data type? ★

Writing

Learn to write data types

Can you write the same number as different types?

Can you write boolean and strings?

Uninformed high-agency (UH) sidebar ↑

Data Types ▾

Overview

↓ Choose any lesson or exercise.

Reading expert

Why we have different data types ✓

Data type examples ✓

Floats vs Integers ✓

Strings: The literal one ✓

Booleans: True or False

You've practiced this concept and related ones already, so this exercise is review.

Do you know the rules about data types?

What's that data type? ★

Writing novice

Learn to write data types

You've practiced related concepts, but not yet this one. Try this exercise for a challenge!

Can you write the same number as different types?

Can you write boolean and strings?

Informed high-agency (IH) sidebar →

Figure 3: Sidebars for the uninformed high-agency (UH) and informed high-agency (IH) variations of Codeitz. The UH version (left) only shows what instruction and exercises a learner has completed (using check marks and stars). The IH version (right) includes skill bars (dotted ovals) to denote estimated mastery and blue goal-oriented recommendations for next exercises to consider (dotted rectangles).

learning experiences. They would select a concept from the world view (Fig. 2), then use the sidebar as shown on the left of Figure 3 to look at what instruction and exercises they had/ had not completed. With this information, learners using the UH version of Codeitz had the freedom to explore any instructional material in any concept.

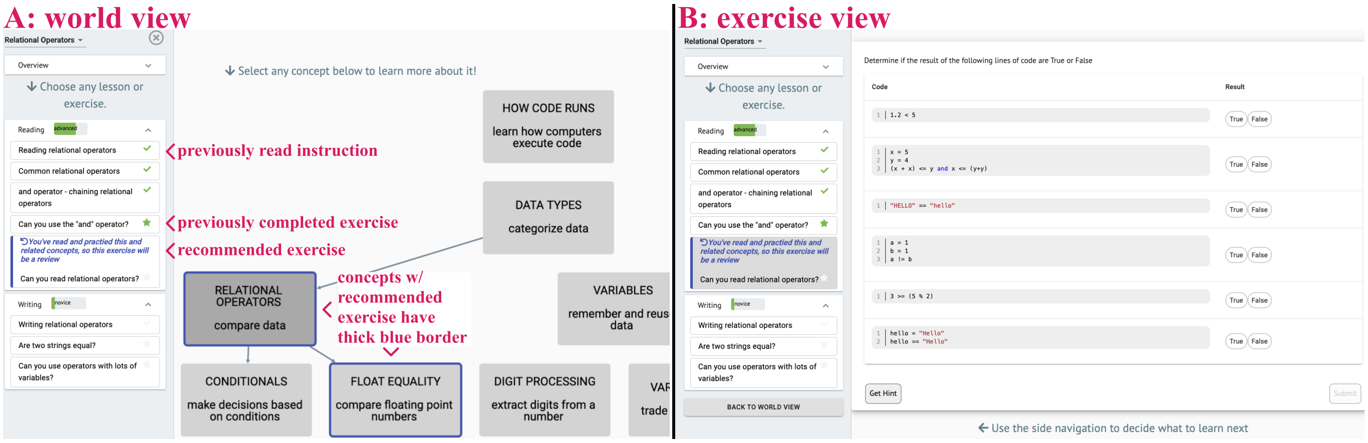
IH: The informed high-agency version provided recommendations and skill bars while requiring learners to exert agency.
Recommendations highlighted specific concepts in the world view and certain exercises in the side bar. The right side of Figure 3 shows the sidebar for the IH version with skill bars to show estimated mastery of a concept and recommendations to show recommended exercises which may support different goals (e.g. review, challenge). We intended for the IH version of Codeitz to reflect a recommender system where learners could follow recommendations but could also choose to deviate from them at no penalty. Figure 4 shows the interactions of the IH condition.

IL: The informed low-agency version provided recommendation and skill bars, but limited choices to a single next recommendation or prior exercises.
We intended for the IL version of Codeitz to reflect a Computerized Adaptive Test (CAT) [9, 10] or basic Intelligent Tutoring System (ITS) [68] where the system decided the next exercise for learners. So rather than being free to choose a concept and then an exercise as high-agency conditions did, learners using the IL version clicked a "next" button and the system selected the concept of the top recommended exercise. From there, they could choose to 1) do the exercise, 2) read related instruction, or 3) review any prior concepts. Only after they attempted the exercise would they be provided with a new one.

Adaptivity with Bayesian Knowledge Tracing (BKT)

To estimate learners' knowledge and recommend/select exercises for IH and IL designs, we implemented a modified version of the Bayesian Knowledge Tracing (BKT) algorithm [16]. BKT is a Hidden Markov Model that has the key assumption that learners can undergo a one-way transition from the *unlearned* to *learned* state for each concept, after which there is a change in the probability they will get an exercise correct [52, 30, 33]. While BKT typically assumes items to be equal, we used the Knowledge Tracing Item Difficulty Effect Model (KT-IDEM, [49]) to encode exercise difficulty.

Our model had two parameters at the concept level and two at the exercise level. The concept-level parameters were $P(L_0)$, the probability a learner already knew a concept before attempting an exercise, and $P(T)$, the probability of a learner transitioning from an unlearned to learned state after an exercise attempt. The exercise-level parameters are $P(S_m)$, the probability of a learner who had learned a concept *slipping* and getting an exercise m wrong, and $P(G_m)$, the probability of a learner who had not learned a concept *guessing* and getting m correct. A more difficult exercise would have a higher slip probability and a lower guess probability. We fitted these model parameters using expert review [36] based on $\approx$15 responses and exercise properties (e.g. closed or open form,


A: world view
Relational Operators
Overview
Choose any lesson or exercise.
Reading
advanced
Reading relational operators
Common relational operators
and operator - chaining relational operators
Can you use the "and" operator?
You've read and practied this and related concepts, so this exercise will be a review
Can you read relational operators?
Writing
novice
Writing relational operators
Are two strings equal?
Can you use operators with lots of variables?
previously read instruction
previously completed exercise
recommended exercise
Select any concept below to learn more about it!
HOW CODE RUNS
learn how computers execute code
DATA TYPES
categorize data
RELATIONAL OPERATORS
compare data
concepts w/ recommended exercise have thick blue border
VARIABLES
remember and reus data
CONDITIONALS
make decisions based on conditions
FLOAT EQUALITY
compare floating point numbers
DIGIT PROCESSING
extract digits from a number
VAR
trade
B: exercise view
BACK TO WORLD VIEW
Code
Result
True
False
Get Hint
Submit
Use the side navigation to decide what to learn next

Determine if the result of the following lines of code are True or False

```
1.2 < 5
```

```
x = 5
y = 4
(x + x) <= y and x <= (y+y)
```

```
"HELLO" == "hello"
```

```
a = 1
b = 1
a != b
```

```
3 >= (5 % 2)
```

```
hello = "Hello"
hello == "Hello"
```

Figure 4: Two primary views of Codeitz for informed high-agency (IH) condition. **A**: The learner followed a the recommendations and selected the *Relational Operators* concept and is able to view the instruction and exercises for that concept in the sidebar. **B**: After clicking on the recommended exercise (*Can you read relational operators?*), the learner is then taken to the exercise view where they can attempt the exercise as practice. In the uninformed high-agency (UH) condition, there are no blue recommendations. In the informed low-agency (IL) condition, there is no world view (A) and learners must instead follow system recommendations.

perceived difficulty of exercises), and knowledge domain; all parameters ranged from 0.01 to 0.25.

Put together, we used this modified version of BKT to estimate the probability of getting an exercise correct. The estimated probability a learner will get a given exercise m correct at the n-th attempt is $P(correct_n|M_n = m) = P(L_n)(1 - P(S_m)) + (1 - P(L_n)(P(G_m))$, or the sum of the probability of getting the exercise correct in the learned and unlearned states. We used this to incrementally update the probability a learner was in the learned state after the n-th opportunity as follows: $P(L_n|correct_{n,m}) = P(L_n)(1 - P(S_m))/P(correct_{n,m})$. We used this probability of being in a learned state as an estimate of a learner’s understanding of that concept.

To select exercises, we used the BKT-Sequence Algorithm [18] which orders exercises based on a minimum difference between predicted difficulty and desired difficulty based on current learner understanding. After each exercise attempt, the probability a concept is learned ($P(L_n)$) was updated. We then updated the sequence of recommended exercises:

1. Calculate $MinScore$ and $MaxScore$, the minimum and maximum $P(correct_m)$ for all incomplete exercises.
2. For all incomplete exercises, calculate $WantedScore_m = (MaxScore - MinScore) \cdot (1 - P(L_n))$ where n is the concept corresponding to each exercise.
3. Calculate $diff_m = WantedScore_m - P(correct_m)$.
4. Order exercises in ascending order by $|diff_m|$.

We then selected the top two exercises, as well as the top two exercises from current, parent, or child concepts.

STUDY: AGENCY ON EXPERIENCES, LEARNING

To understand the effects of varying information and agency afforded in our three versions of Codeitz on engagement and learning, we conduct a between-subjects study with 79 novice programmers. We sought to be ecologically consistent with discretionary use tools to support novice programmers learning in formal learning environments (e.g. an online practice tool used by students in an introductory CS course).

The study included novice programmers who were primarily university and community college students near an industrialized urban center of the United States. We recruited participants through flyers placed throughout a university and surrounding area, pitches to computing-related courses, and posts to closed social media groups. Our inclusion criteria specified participants had to be at least 18 years old, never learned or used Python, completed at most one non-Python programming course prior (although 9 participants violated this criteria: UH:2, IH:5, IL:2), have access to a computer with internet, and be fluent in English. Participants’ self-reported ethnicities were Asian (52%), Caucasian (27%), Hispanic/Latinx (9%), mixed race (6%), and Black/African (3%), with 4% choosing not to disclose. Genders of participants were men (51%), women (44%), and non-binary (1%), with 4% not disclosing. Most (84%) reported working towards one of 40+ different degrees (roughly, physical sciences: 23% of all participants, computer science & informatics: 19%, engineering: 16%, humanities, arts, social sciences: 10%, business & finance: 8%, math: 1%, undeclared: 3%).

Participation in the study began with participants creating an online acount and then getting randomly assigned to one of three conditions. They then completed a pre-survey which asked questions relating to programming self-efficacy (as measured by a programming self-efficacy survey [53]), mindset [20], and motivation for participating in the study. They then used Codeitz across the span of a week and then when they felt ready, took a post-survey and post-test. We compensated participants with a $50 gift card upon completion of an exercise in most concepts and the post-survey.

The post-survey asked learners about their experience using Codeitz (which also served as a distractor task [24]), then administered the hour-long post-test, then measured their programming self-efficacy again, then mindset, and finally asked about demographic information. Demographic information was not asked until the end to avoid stereotype threat [60]. The post-test measured learning outcomes for basic Python knowledge taught in Codeitz, adapting questions from [69, 50, 12].

For questions relating to learner experience, we focused on how learners decided what to learn next and how important different features of Codeitz were in their decision.

We used the following questions to analyze experiences:

1. After you were done with a lesson or exercise in Codeitz, how did you decide what to do next? (open response).
2. Think back to when you finished an exercise. How important were the following parts of Codeitz in deciding what to do next? (Likert-type, shown in Fig. 5).
3. Were there other parts of Codeitz that you considered when deciding what to do next? If so, please describe them and how important they were. (open).
4. If you remember seeing the blue recommendation text (pictured below), how did you use it to decide what to do next? (open).
5. What about using Codeitz caused you to feel frustrated, if anything? (open).
6. What about using Codeitz was helpful to you, if anything? (open).

RESULTS: EXPERIENCES, LEARNING

To answer our research question of the effect of varying levels of agency and information to support agency on engagement and learning outcomes, we analyzed two aspects: 1) learners' experience in the three designs and 2) the outcomes of these experiences on learning.

Experiences varied by condition, performance

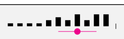
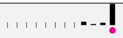
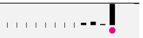
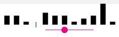
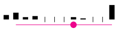
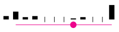
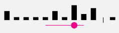
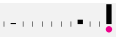
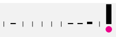
To analyze learners' experiences, we took two perspectives, first analyzing post-survey responses and log data on the use and perception of Codeitz features, and then analyzing learners' experiences between the three design alternatives.

Use and Perception of Agency Information

Figure 5 shows participants' ratings of the importance of design features in Codeitz across conditions (as described in Fig. 1). They rated these features on a five point Likert-type scale from "Not at all important" to "Extremely important." This scale also had a sixth "Not applicable" (N/A) option because some features were not present in some versions of Codeitz.

Qualitative and Likert-type survey responses suggested that the features available in all three conditions were generally viewed as valuable to learning. Participants in all conditions found the **progress indicators** (check marks and stars) denoting instruction and exercise completion to be helpful: of the 79

Table 1: Data by condition. Sample size (n) includes number of low (↓) and high (↑) performers on post-test. Histograms of post-test score (max: 39.5), number of Codeitz exercises attempted (max: 44), and number completed (max: 43) shown with median ($\tilde{x}$) and interquartile range (iqr) (approx. to histogram bin).

cond.	n	test score	# attempted	# completed
IH	25 ↓: 7. ↑:9	 $\tilde{x}$:23.5; iqr:12	 $\tilde{x}$:43; iqr:1	 $\tilde{x}$: 43; iqr: 3
IL	31 ↓:12. ↑:12	 $\tilde{x}$:21.8; iqr:17	 $\tilde{x}$:29; iqr:34	 $\tilde{x}$:29; iqr:34
UH	23 ↓: 7. ↑:5	 $\tilde{x}$:23.0; iqr:14	 $\tilde{x}$:43; iqr:0	 $\tilde{x}$:43; iqr:2.5

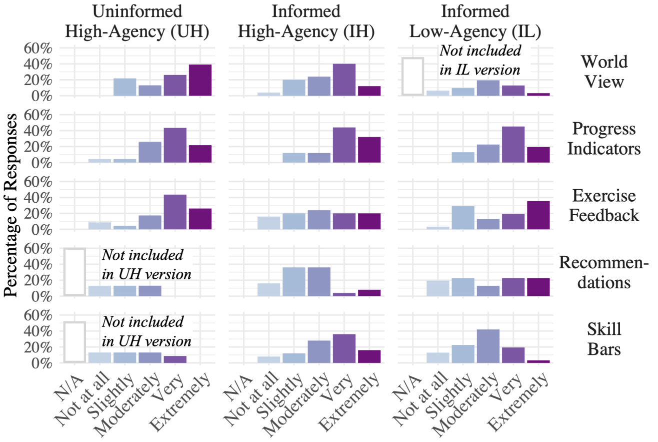


Figure 5: Importance of different features of Codeitz by condition. Not all features were present in each version of Codeitz (see Fig. 1).

participants across all conditions, 89% found the progress indicators at least moderately helpful and 68% found them very or extremely helpful (see second row of Fig. 5). Few, however, reported how they used them. Despite **exercise feedback** in Codeitz consisting only of binary correctness feedback for at least 85% of exercises, participants still found the feedback valuable (third row Figure 5). But participants across all conditions tended to want more intermediate feedback on their exercises to help them persevere after getting incorrect exercise attempts, something that less than 15% of exercises had. P68 in the UH condition reflected this tendency to want intermediate feedback: "*hints and better feedback when you get an answer incorrect... would help me feel more confident about completing the task.*"

IH and UH participants who had the **world view** reported it as important to guiding their learning: 22 of 23 UH participants and 11 of 25 IH participants reported using the world view to decide what to learn next. Most participants in both high-agency conditions (UH: 65%, IH: 52%) reported finding the world view very or extremely important. Many valued the world view for how it explicitly revealed dependencies ("*I like seeing the connection between concepts in the world view. That was very helpful to see how concepts fit together.*", P18, UH). Others noted that "*hidden dependencies*" between concepts caused confusion (e.g., "*the practice in the [Conditionals] has questions that need you to actually read Arithmetic Operators first before you can solve it... I need[ed] to go back to the other concepts before solving the exercise*", P1, IH).

Participants viewed the information provided only to the informed conditions (BKT-based recommendations and skill bars) as less valuable. Only 53% of the 56 (IH & IL) participants who saw them found **recommendations** at least moderately important to deciding what to learn next (Fig. 5). A common complaint was that recommendations tended to "*jump around*" (P72, IL) or "*jump too far*" (P51, IH), suggesting that recommendation behavior was unpredictable. One reason may be because of *cold-start* in which the recommender system initially had no information about a new user and therefore was more prone to making poor recommendations [35]. Recommendations did seem to improve as participants used Codeitz more and the system collected more data (e.g., "*...at first I would just click on any blue exercise without realizing they could require applying concepts that were entirely unfamiliar (this lead to some frustration), but eventually after solely using the [next] button to proceed I would only encounter exercises after I was sure I could complete them.*", P74, IL).

Compared to recommendations, participants found **skill bars** as more valuable, with 63% of participants reporting them to be least moderately important. An IH high-performer reported using skill bars to determine what to learn next: "*I considered when the green skill bars would become 'advanced' which helped me know whether or not I should move on to the next topic.*" (P45, IH). Still, some participants were skeptical of the skill bar ratings: "*The skills levels, 'novice, expert, etc' varied depending on which section I was completing. For example, Even though the end section showed my reading skill as 'expert', if I clicked back to the initial section it showed it as a 'novice'. This made progress tracking feel a little empty, as it appeared to be simply used as visual feedback, not actual tracking*" (P55, IL). This unexpected behavior was because estimates of knowledge were localized to specific concepts and had no relationship to other concepts, even if those other concepts were dependencies (elaborated on later).

Learning Experiences by Condition and Outcomes

To understand how learners' experiences varied by conditions, we analyzed open-ended responses within each condition and compared the responses of participants who scored in the top 1/3 (*high performers*, $> 28.25/39.5$) and bottom 1/3 (*low performers*, < 18.5) on the post-test across all study participants; following the *contrasting groups* method of psychometrics [36, 37]. We conducted an open coding and thematic analysis [1] of post-survey responses from these contrasting groups, seeking to understand how they used features of Codeitz to decide what to learn next and what factors may help explain their performance on the post-test. We used log data (e.g. number of exercises attempted) to triangulate our findings.

Participants in the **Uninformed High-Agency (UH)** condition (N=23) tried to define their own learning trajectory to varying success. Without recommendations or skill bars, UH participants had to decide what to learn next using the world view showing them the concept hierarchy, progress indicators showing them what instructions and exercises they've completed (check marks for read instructions, stars for correct exercises), and exercise correctness feedback. Almost all (22) of the 23 UH participants explicitly mentioned using the world view to decide "*I looked at the flowchart and picked a connecting branch.*" (P57). Overall, participants in the UH condition tended to attempt all the exercises, with 78% attempting all the exercises and 65% getting them all correct (all participants were allowed unlimited retries).

High performers in the UH condition all reported exerting agency. While the UH condition did not have recommendations or skill bars to inform learners, some of the 5 high-performers in this condition noted still being able to deviate from the world view's explicit paths in ways that benefited their learning. For example, 2 high-performers noted trying exercises and then reviewing lessons if they were unfamiliar with the exercise. Another skipped to trying exercises first but jumped back to reading lessons when necessary: "*I mostly skipped around to the exercises, if I felt like I could understand what was going on in the lesson, and then moved back through the lesson if I couldn't do an exercise.*" (P5). The remaining 3 high-performers reported finding Codeitz the structure and presentation of the curriculum helpful, the design of the curriculum as intuitive: "*The lessons were easy to understand and exercises helped cement the knowledge*" (P21).

In contrast to UH high performers, the 7 **UH low performers** noted struggling to navigate their learning experience. For example, P39 was frustrated because there was *"no proper path"* and P63 felt "*the order [of concepts] did not seem intuitive.*" P18, who reported minimal programming self-efficacy prior to the study, noted how his confidence affected what he chose to learn next: "*To decide which lesson I would try next from a lower level, particularly the second level, I* *looked at which concept I felt most confident taking on first.* (P18). Low-performing UH participants also noted wanting additional instructional content, such as "*video explanation*" (P3) and more feedback in code writing exercises (P9).

Participants in the **Informed High-Agency (IH)** condition (N=25) reported similar experiences to the UH condition, with additional comments about the recommendations and skill bars. To decide what to learn next, 11 of the 25 IH participants reported using the world view, 10 reported following the recommendations, and 2 reported trying the recommendations but then abandoning them. So whereas almost all UH participants reported following the world view, less than half of the IH participants reported doing so, with 28% reporting at least trying to follow the recommendations. Overall, participants in the IH condition also tended to attempt all the exercises, with 64% attempting all the exercises and 56% of IH participants getting them all the correct.

IH high performers reported evolving interpretations of the recommendations as they deviated from the world view's prescribed paths. Three of the 9 high-performing IH participants reported using the recommendations; of those 3, 2 of them reporting trying the recommendations at first, but then abandoning them because they led them to exercises that were too advanced: "*At first, I looked at the blue highlighted boxes. However, I felt like it made me jump too far. For example, one lesson had started talking about if statements but I hadn't learned the syntax for those yet. So then I just followed the tree from top down, left to right.*" (P25). Three high-performing IH participants ignored the recommendations because they wanted to complete all of the curriculum. P1 noted how he ignored the recommendation but how he could see its benefit for less motivated learners: "*I did not really pay attention to the blue recommendation because I was motivated to do all the lessons and the practices to receive maximum knowledge. I think this blue recommendation might be useful for people who have low motivation to do more exercises.*" Log data confirmed tendency to do all exercises, as all high-performing IH participants completed at least 40/43 of the exercises (93%), although this complete coverage of exercises was consistent across the entire IH condition.

In contrast to the high-performing IH participants, the 7 **low-performing IH** participants tended to ignore or misinterpret the recommendations and followed a perceived intended path. Three participants ignored the recommendations, such as P24 who "*usually just did the problem even though it was review.*" Another participant was confused by the recommendations updating: "*... sometimes [a recommendation] would be there and sometimes it would not. Typically I would see this after I finished in exercise.*" (P38). Of the two participants who reported using the recommendations, P43 reported that he "*did what [the recommendation] said*" while P23 used the recommendation to estimate how much time an exercise would take: "*While having limited free time, it was helpful to see a note indicating that the next tab was a review exercise, meaning it would likely be quick to complete.*" (P23). Three low performing IH participants struggled with having to choose their own trajectory. P38 reflected this in her description of how she got lost and found choosing what to learn next frustrating: "*I found the layout of what lessons to take were confusing. I went from top to bottom and left to right, however, during the exercises I would find myself lost on multiple occasions. This would be due to either skipping sections but having to use it before I learned the material. Instead of having the choice of choosing what lesson to take next, it would have been more helpful if it was just given* (P38).

Participants in the third and final **Informed Low-Agency (IL)** condition (N=25) only had three choices at any given time: completing the given exercise, reviewing instruction related to the exercise, or reviewing prior lessons and exercises. Three participants found this lack of overview made it challenging to keep track of how much they had completed, how much remained, and how the concepts related to each other. P72 reported his challenges of keeping track of where he was in his learning process: "*Everything seemed to jump around and it was hard to keep track of what I was on or what I was supposed to do next.*" (P72). Overall, participants in the IL condition tended to attempt fewer exercises than the high-agency conditions: whereas both high-agency conditions had most (78% for UH, 64% for IH) participants attempt all 43 exercises, only 41% of participants in the IL condition attempted all the exercises. All 41% of those participants did get all exercises correct, though.

The 12 **high-performing IL** participants varied in how they interpreted the next exercise presented to them. Half (6) reported viewing recommendations as indicators of an exercise or a required next step: *When I saw this blue recommendation, I would make sure to click it in order to complete it as it seemed to mean 'required'* (P42). Three others reported using the recommendation text as informative in deciding whether to attempt an exercise, read instruction, or go back to a previous exercise: *I only used it to see how difficult the exercise was. I would still go straight to the exercise even it told me I hadn't learned about the concept, and I will come back to it later if I didn't figure it out. If it told me it's something I had already learned, I wouldn't leave the exercise until I figured it out.* (P35). The remaining 3 reported not using recommendations (1) or did not comment on their usage (2). Multiple participants reported a desire to have an overview of all concepts and explore concepts more freely: "I didn't know how many topic there are in total and could only view them after doing the previous topic and unlocking it. I feel it would be better if I can see how much I am completing and how much still has to be done" (P75).

In contrast to the high-performing IL participants, the 12 **low-performing IL** participants reported relying much less on the recommendations. Three reported not even seeing or noticing the recommendations. Of the 4 low-performing IL participants that mentioned using recommendations, two saw them as indicators of "*a signal that the selected block [was] an exercise*" (P61). One participant "*used [the recommendation] as an indicator for a concept practice/challenge*" and that "*the practice challenges were very helpful... in learning python*" (P61). Low-performing IL participants also completed fewer exercises: Only 25% (4) of these participants attempted more

Table 2: Coefficients of linear regression to model learning outcomes (post-test scores). *** indicates $p < 0.001$, * that $p < 0.05$, & . that $p < 0.10$.

coefficient	estimate (std. err.)	t	Pr(>|t|)
(Intercept)	14.41 (2.72)	5.301	0.000 ***
condition: IL	0.70 (2.58)	0.270	0.787
condition: IH	0.90 (2.74)	0.328	0.744
self-efficacy (pre)	2.43 (1.02)	2.387	0.020 *
taken CS course	4.62 (2.52)	1.836	0.070 .

Table 3: ANOVA results and effect sizes for linear regression of post-test scores. ε denotes a small positive value (0.001 – 0.004). * indicates that $p < 0.05$.

variable (df)	SE	F	Pr(>F)	η^2 [95% C.I.]
condition (2)	45	0.3	0.776	0.006 [0, 0.08]
self-eff, pre (1)	499	5.7	0.020 *	0.066 [ε, 0.21]
taken CS course (1)	583	6.7	0.012 *	0.077 [ε, 0.23]
residuals (74)	6483			

than half of the exercises; in contrast, 81% (9 of 11) high-performing IL participants completed more than half of the exercises.

Learning & Exercise Completion by Condition

Condition, Self-efficacy, & Prior Knowledge on Post-Test

To understand how the varying designs of Codeitz conditions affected learning, we used a linear regression to model post-test scores. In addition to passing into the regression the condition participants were in (UH, IH, IL), we also considered self-efficacy prior to using Codeitz (range: 1-7) and whether a participant reported taking a prior CS/programming course (true/false), as both self-efficacy and prior knowledge are important to learning [22]. We found no violations of linear regression assumptions: normality (Shapiro-Wilk, p=0.23), homoscedacity (spread-location plot), and linearity [6, 29, 54]. Table 2 shows the coefficients of the linear regression.

Table 3 shows the results of a linear regression model analysis of variance (ANOVA). The ANOVA indicated a statistically significant effect on post-test scores of prior self-efficacy ($F(1,74) = 5.7, p < 0.05$). Whether a participant had previously taken a programming course was also had a statistically significant effect ($F(1,74) = 6.7, p < 0.05$). Both significant factors had medium effect sizes ($\eta^2 > 0.06$) with large confidence intervals which did not include zero. The condition participants were in did not have a statistically significant effect ($F(2,74) = 0.3, n.s.$).

We conducted non-parametric post-hoc analyses to understand how prior self-efficacy and programming course experience affected post-test score. The median post-test score of participants who had taken a prior programming course was 29.12 (IQR = 13.4) and of participants who had not was 21.50 (IQR = 15.3). This difference was statistically significant according to a Mann-Whitney U test ($U = 364.5$, $p = 0.012 < 0.05$). We interpreted the medium Vargha and Delaney A effect size to state that there is 69.1% chance a post-test score for a random participant who has taken a programming course will be greater than a score for a random participant who has not [65]. For self-efficacy, we calculated Kendall's non-parametric rank correlation [31]. We found a significant correlation ($\tau = 0.25, p = 0.0014 < 0.01$) between prior programming self-efficacy and post-test score. We convert τ to $r = 0.38$ [66, 32] and identified a medium effect size between prior self-efficacy and post-test score [54].

Number of Exercises Completed by Condition

To check for a difference in the number of exercises completed by condition, we conducted a Kruskal-Wallis test [54]. We decided on this non-parametric test because the data was not normal (Shaprio-Wilks: $p < 0.05$). Table 1 shows the distribution, median, and IQR for the number of completed questions by condition. We found statistically significant differences in number of completed exercises between conditions ($\chi^2(2, N = 79) = 11.33, p = 0.003 < 0.01$.

We conducted a pairwise post-hoc analysis with Mann-Whitney U tests with Holm correction. We found that a statistically significant difference between the IL condition and the other two conditions (Mann-Whitney U for IL/UH: $U = 504, p = 0.014 < 0.05$; IL/IH: $U = 225, p = 0.014 < 0.05$). We can interpret the medium Vargha and Delaney A effect sizes to say that there is a 71% chance that a random UH participant completed more Codeitz exercises than a random IL participant, and that there is a 71% a random IH participant completed more exercises than a random IL participant.

DISCUSSION: INTERPRETATIONS & IMPLICATIONS

The objective of this study was to jointly understand how affording and informing agency affected engagement and learning outcomes. We did so by designing three variations of a self-directed online learning environment that varied the amount of agency or information afforded to participants as they learned introductory Python.

We found that the specific features offered in these three conditions led to very different learning experiences and degrees of engagement, but that these differing experiences led to no detectable effect on learning outcomes. We also found that low-agency (IL) participants completed significantly fewer exercises than high-agency (IH, IL) ones.

There are multiple ways to interpret these findings related to learner experience and learning outcomes. One interpretation is that our recommendations were not "intelligent" enough to be helpful. Our BKT implementation faced challenges such as parameter tuning [27] and cold-start [35], as consistent with most statistical models. While we did our best to fit parameters according to best practices and given the response data we had available, we also recognized that better parameters could improve the performance of BKT. But Codeitz is representative of a discretionary use self-directed online learning environment in that recommendations and item selection will never be perfect or optimal for all learners, especially early on before we have a large corpus of response data. So understanding how to design information such as adaptive recommendations to affect agency and learning also requires understanding how learners interpret and use information that come with the inevitable imperfections and inaccuracies of data-driven adaptation. And despite many participants feeling like the recommendations "jumped around" and were not always accurate, participants in both high and low agency conditions still found ways to use them to inform their decision-making process and learning. So while our BKT model had identified problems, participants could still use information from it. And participants' experiences and reports can help better inform how we design adaptive online environments that promote learner agency.

Another interpretation is that other confounds made our post-test an invalid or unreliable measure of learning. Because we wanted to investigate the design of self-directed online learning, we set up our study such that it could emulate this discretionary, informal learning. We did so by having participants learn on their own time across the span of a week and then take the post-test whenever they felt ready. While this experimental design introduced confounds including variation in amount of time spent learning and an uncontrolled test-taking environment, they were externally valid to many online learning environments (e.g. MOOCs, online coding platforms, remote/hybrid courses). Such confounds were also distributed across the conditions. Furthermore, post-test items came from concept inventories [50] or were piloted with representative users with think-aloud [21].

Design Considerations and Future Work

A third interpretation of our findings is that designing for agency is nuanced and requires careful design considerations we are only beginning to understand. While prior work investigated varying agency to measure its effect on learning, we designed and varied the information and agency afforded to learners. Our results suggest possible explanations and design considerations to explore in future work:

The value of agency may be dependent on the structure of knowledge to learn. Relating to programming, this study used learning materials with concepts that have rigid hierarchical relationships. This knowledge domain may lend itself to more linear instructional content. Agency to support learning here may be in the form of jumping ahead for a challenge or back for review. In contrast, learning to use programming for expression (e.g. with Scratch) may lend itself more to non-linear instruction. Agency in this case may be in the form or exploration of one of many paths. Therefore, how to afford agency may be dependent on the structure of the knowledge domains and learning objectives.

Agency may be valuable to more than just learning outcomes. Our findings suggest that agency may support motivation to continue learning. Prior work has generally found more agency relating to increased motivation (e.g. [57, 17, 56, 47, 25]). Our findings suggest that there was a 71% chance a random high-agency participant completed more exercises than a low-agency one. This suggested that affording learners the agency to see everything there was to learn (with the World View) and choose for themselves may have had a motivational benefit to help learners continue to engage.

Recommendations may have different roles to different learners. In Codeitz, we intended for the recommendations (Fig. 1d) to be cues to exert agency. While learners in the informed high-agency condition tried to use the recommendations to guide them, many treated the recommendations not as cues as to what to expect from a given exercise, but simply as indicators of an incomplete exercise. It may be important to consider not only the intended role of cues to inform agency, but also to consider alternatives ways learners may interpret them initially as well as after some interactions.

Consider learners' prior experiences with related tools. Just as learners come with prior perceptions related to what they are learning, they also come with prior perceptions related to how to interact with learning environments. While we designed Codeitz to not have an apparent order in high-agency conditions, we found that a majority of participants (in the UH condition especially) reported following or trying to follow an intended order. Such behavior might prevent any potential benefits from exercising agency from materializing.

Overviews, while valuable, may indirectly constrain learners' decisions. Learners found value in Codeitz's world view, suggesting it provided an integrated view of concepts to be learned. But the overview might have also acted as an *indirect control* [59], limiting agency. Designs may need to consider the unintended side effects of offering conceptual overviews on how learners choose to sequence their learning.

Our evidence, and these possible interpretations, suggest that designing for agency, and in particular, designing information that encourages agency, is far from straightforward. Just as offering choice is not consistently beneficial to learners, offering information to support those choices is not consistently beneficial either. Future work should explore with more granularity the interaction between self-directed learning environments, learners' interpretation of what the environment provides, and learning outcomes. And designers should be wary about the benefits of learner agency, and pay close attention to the specific domain of learning and the specific unintended side effects of how learners use the affordances in a self-directed learning environment.

ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 1735123, 1539179, 1703304, 1836813, 1450681, and 12566082 and unrestricted gifts from Microsoft, Adobe, and Google. We thank Alannah Oleson for the Gender Mag walk-through [7], and our Code & Cognition lab mates for their constructive critique. Supplemental material (code, surveys, post-test) can be found at https://github.com/codeandcognition/archive-2020las-xie.

REFERENCES

[1] Anne Adams, Peter Lunt, and Paul Cairns. 2008. A qualititative approach to HCI research. In *Research Methods for Human-Computer Interaction*, Paul Cairns and Anna Cox (Eds.). Cambridge University Press, A qualititative approach to HCI research.

[2] Richard C Atkinson. 1972. Optimizing the learning of a second-language vocabulary. *J. Exp. Psychol.* 96, 1 (Nov. 1972), 124–129.

[3] Albert Bandura. 2001. Social Cognitive Theory: An Agentic Perspective. *Annu. Rev. Psychol.* 52, 1 (2001), 1–26.

[4] Albert Bandura. 2006. Toward a Psychology of Human Agency. *Perspect. Psychol. Sci.* 1, 2 (June 2006), 164–180.

[5] James R Bettman, Mary Frances Luce, and John W Payne. 1998. Constructive Consumer Choice Processes. *J. Consum. Res.* 25, 3 (Dec. 1998), 187–217.

[6] Peter Bruce and Andrew Bruce. 2017. *Practical Statistics for Data Scientists: 50 Essential Concepts*. "O'Reilly Media, Inc.".

[7] Margaret Burnett, Anicia Peters, Charles Hill, and Noha Elarief. 2016. Finding Gender-Inclusiveness Software Issues with GenderMag: A Field Investigation. ACM Press, 2586–2598.

[8] Sandra L Calvert, Bonnie L Strong, and Lizann Gallagher. 2005. Control as an Engagement Feature for Young Children's Attention to and Learning of Computer Content. *Am. Behav. Sci.* 48, 5 (Jan. 2005), 578–589.

[9] Hua-Hua Chang. 2004. Understanding Computerized Adaptive Testing. *The Sage handbook of quantitative methods for the social sciences* (2004), 117–133.

[10] Hua-Hua Chang. 2015. Psychometrics behind Computerized Adaptive Testing. *Psychometrika* 80, 1 (March 2015), 1–20.

[11] Xingliang Chen, Antonija Mitrovic, and Moffat Mathews. 2019. Investigating the Effect of Agency on Learning from Worked Examples, Erroneous Examples and Problem Solving. *International Journal of Artificial Intelligence in Education* (2019).

[12] Michael Clancy and Marcia C Linn. 1992. *Designing Pascal Solutions: A Case Study Approach*. Computer Science Press.

[13] Gemma Corbalan, Liesbeth Kester, and Jeroen J G van Merriënboer. 2008. Selecting learning tasks: Effects of adaptation and shared control on learning efficiency and task involvement. *Contemp. Educ. Psychol.* 33, 4 (2008), 733–756.

[14] Gemma Corbalan, Liesbeth Kester, and Jeroen J G van Merriënboer. 2009. Combining shared control with variability over surface features: Effects on transfer test performance and task involvement. *Comput. Human Behav.* 25, 2 (2009), 290–298.

[15] Gemma Corbalan, Liesbeth Kester, and Jeroen J G van Merriënboer. 2011. Learner-controlled selection of tasks with different surface and structural features: Effects on transfer and efficiency. *Comput. Human Behav.* 27, 1 (Jan. 2011), 76–81.

[16] Albert T Corbett and John R Anderson. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Model. User-adapt Interact.* 4, 4 (Dec. 1994), 253–278.

[17] Diana I Cordova and Mark R Lepper. 1996. Intrinsic motivation and the process of learning: Beneficial effects of contextualization, personalization, and choice. *J. Educ. Psychol.* (1996).

[18] Yossi Ben David, Avi Segal, and Ya'akov (kobi) Gal. 2016. Sequencing educational content in classrooms using Bayesian knowledge tracing. In *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge*. ACM, 354–363.

[19] Steven P Dow. 2008. *Understanding User Engagement in Immersive and Interactive Stories*. Ph.D. Dissertation. Georgia Institute of Technology.

[20] Carol S Dweck. 2008. *Mindset: The New Psychology of Success*. Ballantine Books.

[21] Karl Anders Ericsson and Herbert Alexander Simon. 1993. *Protocol Analysis: Verbal Reports as Data Revised Edition*. The MIT Press.

[22] Sally A Fincher and Anthony V Robins. 2019. *The Cambridge Handbook of Computing Education Research*. Cambridge University Press.

[23] Terri Flowerday and Gregory Schraw. 2000. Teacher Beliefs about Instructional Choice: A Phenomenological Study. *Journal of Educational Psychology* 92, 4 (Dec. 2000), 634–645. DOI: http://dx.doi.org/10.1037/0022-0663.92.4.634

[24] Peter A Frensch, Axel Buchner, and Jennifer Lin. 1994. Implicit Learning of Unique and Ambiguous Serial Transitions in the Presence and Absence of a Distractor Task. *J. Exp. Psychol. Learn. Mem. Cogn.* 20, 3 (1994), 567–584.

[25] John P Fry. 1972. Interactive relationship between inquisitiveness and student control of instruction. *J. Educ. Psychol.* 63, 5 (Oct. 1972), 459–465.

[26] Erik Harpstead, J Elizabeth Richey, Huy Nguyen, and Bruce M McLaren. 2019. Exploring the Subtleties of Agency and Indirect Control in Digital Learning Games. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge (LAK '19)*. ACM.

[27] William J Hawkins, Neil T Heffernan, and Ryan S J D Baker. 2014. Learning Bayesian Knowledge Tracing Parameters with a Knowledge Heuristic and Empirical Probabilities. In *Intelligent Tutoring Systems*. Springer International Publishing, 150–155.

[28] Roya Hosseini, I-Han Hsiao, Julio Guerra, and Peter Brusilovsky. 2015. What Should I Do Next? Adaptive Sequencing in the Context of Open Social Student Modeling. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*,

Gráinne Conole, Tomaž Klobučar, Christoph Rensing, Johannes Konert, and Elise Lavoué (Eds.). Lecture Notes in Computer Science, Vol. 9307. Springer International Publishing, Cham, 155–168. DOI: http://dx.doi.org/10.1007/978-3-319-24258-3_12

[29] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. *An Introduction to Statistical Learning* (8 ed.). Springer Texts in Statistics, Vol. 103. Springer New York, New York, NY.

[30] Jussi Kasurinen and Uolevi Nikula. 2009. Estimating Programming Knowledge with Bayesian Knowledge Tracing. In *Proceedings of the 14th Annual ACM SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE '09)*. ACM, New York, NY, USA, 313–317.

[31] Maurice G Kendall. 1938. A New Measure of Rank Correlation. *Biometrika* 30, 1/2 (1938), 81–93.

[32] Maurice G Kendall. 1970. *Rank Correlation Methods*. Griffin.

[33] Mohammad M Khajah, Yun Huang, José P González-Brenes, Michael C Mozer, and Peter Brusilovsky. 2014. Integrating knowledge tracing and item response theory: A tale of two frameworks. In *CEUR Workshop Proceedings*, Vol. 1181. University of Pittsburgh, 7–15.

[34] Sarah Lichtenstein and Paul Slovic. 2006. *The Construction of Preference*. Cambridge University Press.

[35] Blerina Lika, Kostas Kolomvatsos, and Stathes Hadjiefthymiades. 2014. Facing the cold start problem in recommender systems. *Expert Syst. Appl.* 41, 4, Part 2 (March 2014), 2065–2073.

[36] Samuel A Livingston and Michael J Zieky. 1982. *Passing Scores: A Manual for Setting Standards of Performance on Educational and Occupational Tests*. Educational Testing Service.

[37] Samuel A Livingston and Michael J Zieky. 1989. A Comparative Study of Standard-Setting Methods. *Applied Measurement in Education* 2, 2 (April 1989), 121–141.

[38] Dastyni Loksa, Benjamin Xie, Harrison Kwik, and Amy J Ko. 2020. Investigating Novices' In Situ Reflections on Their Programming Process. In *Proceedings of the ACM Technical Symposium on Computer Science Education (SIGCSE), Research Track*. ACM.

[39] Yanjin Long and Vincent Aleven. 2013. Active learners: Redesigning an intelligent tutoring system to support self-regulated learning. In *European Conference on Technology Enhanced Learning*. Springer, 490–495. DOI:http://dx.doi.org/10.1007/978-3-642-40814-4_44

[40] Yanjin Long and Vincent Aleven. 2014. Gamification of Joint Student/System Control over Problem Selection in a Linear Equation Tutor. In *Proceedings of the 12th International Conference on Intelligent Tutoring Systems, ITS 2014*. 378–387. DOI: http://dx.doi.org/10.1007/978-3-319-07221-0_47

[41] Monika Mandl, Alexander Felfernig, Erich Teppan, and Monika Schubert. 2011. Consumer decision making in knowledge-based recommendation. *J. Intell. Inf. Syst.* 37, 1 (Aug. 2011), 1–22.

[42] Sarah Mercer. 2012a. The Complexity of Learner Agency. *Apples - Journal of Applied Language Studies* (2012).

[43] Sarah Mercer. 2012b. The Complexity of Learner Agency. *Apples - Journal of Applied Language Studies* (2012).

[44] Janet Metcalfe, Teal S Eich, and David B Miele. 2013. Metacognition of agency: proximal action and distal outcome. *Exp. Brain Res.* 229, 3 (Sept. 2013), 485–496.

[45] Janet Metcalfe and Herbert S Terrace. 2013. *Agency and Joint Attention*. OUP USA.

[46] National Academies of Sciences, Engineering, and Medicine. 2018. *How People Learn II: Learners, Contexts, and Cultures*. National Academies Press, Washington, D.C.

[47] Thomas O Nelson, John Dunlosky, Aurora Graf, and Louis Narens. 1994. Utilization of Metacognitive Judgments in the Allocation of Study During Multitrial Learning. *American Psychological Society* 5, 4 (1994).

[48] Huy Nguyen, Erik Harpstead, Yeyu Wang, and Bruce M McLaren. 2018. Student Agency and Game-Based Learning: A Study Comparing Low and High Agency. In *Artificial Intelligence in Education*. Springer International Publishing, 338–351.

[49] Zachary A Pardos and Neil T Heffernan. 2011. KT-IDEM: Introducing Item Difficulty to the Knowledge Tracing Model. In *User Modeling, Adaption and Personalization*. Springer Berlin Heidelberg, 243–254.

[50] Miranda C Parker, Mark Guzdial, and Shelly Engleman. 2016. Replication, Validation, and Use of a Language Independent CS1 Knowledge Assessment. In *Proceedings of the 2016 ACM Conference on International Computing Education Research (ICER '16)*. ACM, New York, NY, USA, 93–101.

[51] John W Payne, James R Bettman, and Eric J Johnson. 1993. *The Adaptive Decision Maker*. Cambridge University Press.

[52] Radek Pelánek. 2017. Bayesian knowledge tracing, logistic models, and beyond: an overview of learner modeling techniques. *User Model. User-adapt Interact.* 27, 3-5 (2017), 313–350.

[53] Vennila Ramalingam and Susan Wiedenbeck. 1998. Development and Validation of Scores on a Computer Programming Self-Efficacy Scale and Group Analyses of Novice Programmer Self-Efficacy. *Journal of Educational Computing Research* 19, 4 (1998), 367–381.

[54] Judy Robertson and Maurits Kaptein. 2016. *Modern Statistical Methods for HCI*. Springer.

[55] Ido Roll, Eliane Stampfer Wiese, Yanjin Long, Vincent Aleven, and Kenneth R. Koedinger. 2014. Tutoring Self-and Co-Regulation with Intelligent Tutoring Systems to Help Students Acquire Better Learning Skills. *Design Recommendations for Intelligent Tutoring Systems - Volume 2: Instructional Management* 2 (2014), 169–182.

[56] Jonathan P Rowe, Lucy R Shores, Bradford W Mott, James C Lester, and North Carolina. 2011. Integrating Learning, Problem Solving, and Engagement in Narrative-Centered Learning Environments. *International Journal of Artificial Intelligence in Education* 21 (2011), 115–133.

[57] Richard M Ryan, C Scott Rigby, and Andrew Przybylski. 2006. The Motivational Pull of Video Games: A Self-Determination Theory Approach. *Motiv. Emot.* 30, 4 (Dec. 2006), 344–360.

[58] Robert Sawyer, Andy Smith, Jonathan Rowe, Roger Azevedo, and James Lester. 2017. Is More Agency Better? The Impact of Student Agency on Game-Based Learning. In *Artificial Intelligence in Education*. Springer International Publishing, 335–346.

[59] Jesse Schell. 2014. *The Art of Game Design: A Book of Lenses, Second Edition*. A K Peters/CRC Press. 1–489 pages.

[60] Toni Schmader and Michael Johns. 2003. Converging evidence that stereotype threat reduces working memory capacity. *J. Pers. Soc. Psychol.* 85, 3 (2003), 440–452.

[61] Herbert A Simon. 1955. A Behavioral Model of Rational Choice. *Q. J. Econ.* 69, 1 (Feb. 1955), 99–118.

[62] Erica L. Snow, Laura K. Allen, Matthew E. Jacovina, and Danielle S. McNamara. 2015. Does Agency Matter?: Exploring the Impact of Controlled Behaviors within a Game-Based Environment. *Computers & Education* 82 (March 2015), 378–392. DOI: http://dx.doi.org/10.1016/j.compedu.2014.12.011

[63] Huib K Tabbers and Bastiaan de Koeijer. 2010. Learner control in animated multimedia instructions. *Instructional Science* 38, 5 (Sept. 2010), 441–453.

[64] Michelle Taub, Robert Sawyer, Andy Smith, Jonathan Rowe, Roger Azevedo, and James Lester. 2020. The agency effect: The impact of student agency on learning, emotions, and problem-solving behaviors in a game-based learning environment. *Comput. Educ.* 147 (April 2020), 103781.

[65] András Vargha and Harold D Delaney. 2000. A Critique and Improvement of the CL Common Language Effect Size Statistics of McGraw and Wong. *J. Educ. Behav. Stat.* 25, 2 (June 2000), 101–132.

[66] David A Walker. 2003. JMASM9: converting Kendall's tau for correlational or meta-analytic analyses. *J. Mod. Appl. Stat. Methods* (2003).

[67] Noah Wardrip-Fruin, Michael Mateas, Steven Dow, and Serdar Sali. 2009. Agency Reconsidered. *DiGRA Conference* (2009).

[68] Beverly Park Woolf. 2009. *Building intelligent interactive tutors: student-centered strategies for revolutionizing e-learning*. Morgan Kaufmann Publishers/Elsevier, Amsterdam ; Boston.

[69] Benjamin Xie, Matthew J Davidson, Min Li, and Amy J Ko. 2019a. An Item Response Theory Evaluation of a Language-Independent CS1 Knowledge Assessment. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. ACM.

[70] Benjamin Xie, Dastyni Loksa, Greg L Nelson, Matthew J Davidson, Dongsheng Dong, Harrison Kwik, Alex Hui Tan, Leanne Hwa, Min Li, and Amy J Ko. 2019b. A theory of instruction for introductory programming skills. *Computer Science Education* (Jan. 2019), 1–49.

[71] Benjamin Xie, Greg L Nelson, and Amy J Ko. 2018. An Explicit Strategy to Scaffold Novice Program Tracing. In *2018 ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE '18)*. ACM, New York, NY, USA.