Beetrap-MC: A Minecraft-Based AI Literacy Tool for Teaching Filter Bubbles to Middle School Students

Erfan Farhadi, Kenneth Fei, Yifan Jiang, Zhen Bai
Department of Computer Science, University of Rochester
Rochester, NY, USA
efarhadi@ur.rochester.edu, kfei@u.rochester.edu, yjiang54@ur.rochester.edu, zhen.bai@rochester.edu

Abstract

Recommendation systems shape much of what people, including youth, encounter online, influencing their exposure to information and ideas. Understanding their workings and potential downsides, such as filter bubbles, is increasingly important. At the same time, Minecraft remains one of the most popular and accessible game platforms among students worldwide, making it a promising medium for AI literacy outreach. Building on a previous in-person augmented reality application called BeeTrap, in which players, acting as bees, pollinate flowers and see how similar choices narrow their environment, we created BeeTrap-MC, a Minecraft-based version aimed at broader reach and accessibility. Unlike the original facilitator-led group workshop, BeeTrap-MC is designed for students individual playthrough. We conducted a study with nine middle school participants, using pre-/post-assessments and qualitative interviews to evaluate its effectiveness. Results showed significant learning gains in key AI concepts, such as understanding filter bubbles and their consequences. We also discuss key differences in design, usability and outcomes between BeeTrap-MC and the original, reflecting on trade-offs in adapting a group-based embodied experience into a shorter, self-guided digital format.

Project Page —
inter-play-lab.github.io/beetrap-fabricmc-1.21.4

Introduction

Artificial intelligence (AI) systems increasingly mediate the information people encounter online, influencing opinions, preferences, and decision-making (Flaxman, Goel, and Rao 2016). Among these, AI-based recommendation systems play a pivotal role by personalizing content based on a user's past behavior. While personalization can enhance convenience, it can also inadvertently narrow exposure to diverse information, creating so-called "filter bubbles" (Pariser 2011). This effect is of particular concern for young people, whose media diets and worldviews are still developing (Bulger and Davison 2018). As such, fostering AI literacy in K–12 learners, including understanding the mechanics and societal implications of recommendation systems, is an important educational goal (Touretzky, Gardner-McCune, and Seehorn 2022).

Game-based learning offers an engaging avenue for AI literacy education, providing interactive contexts that make abstract computational concepts more tangible (Gee 2003). Games can serve as embodied metaphors for complex systems, allowing players to explore cause-and-effect relationships through play (Schank and Cleary 2014). Minecraft, in particular, is one of the most widely used game platforms in K–12 education worldwide due to its accessibility, creative affordances, and active educational community (Karsenti and Bugmann 2019). Prior work has leveraged Minecraft for teaching topics from computational thinking to environmental science (Short 2012), but its potential for AI literacy outreach remains underexplored.

The original *BeeTrap* application (Zhou 2024) was developed based on the bee pollination analogy for recommendation system as an in-person, augmented reality (AR) learning experience in which players, acting as bees, pollinate flowers and observe how repeated, similar choices lead to reduced diversity in their environment. This embodied activity aimed to help students understand how algorithmic personalization can limit exposure to diverse content. The original BeeTrap's embodied, analogy-based AR design improved students' understanding of AI recommendation systems and encouraged critical reflection on filter bubbles. However, as a facilitator-led group activity, it required up to two hours of class time, specialized equipment, and instructor support, which limited the scalability of the learning experience.

In this work, we present *BeeTrap-MC*, a Minecraft-based adaptation of BeeTrap designed for broader reach and accessibility. BeeTrap-MC maintains the pollination analogy but reimagines it as a self-guided, 30-minute individual playthrough that can be run on widely available PC or laptop devices. We evaluated BeeTrap-MC with nine middle school students during a summer program, using pre-/post-assessments and qualitative interviews to measure its effectiveness. Our results show learning gains in key AI concepts, particularly in understanding filter bubbles and their consequences. We also reflect on the trade-offs of adapting a group-based, embodied AR activity into a shorter, individual digital format, contributing insights for scaling AI literacy resources in K–12 contexts.

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Background and Related Work

AI Literacy in K–12 Education

AI literacy refers to the knowledge, skills, and dispositions that enable individuals to critically understand, use, and evaluate AI systems (Touretzky, Gardner-McCune, and Seehorn 2022). In K–12 education, developing AI literacy is increasingly seen as essential given the ubiquity of AI in daily life and its societal impact (Bulger and Davison 2018; Casal-Otero et al. 2023). Recent years have seen the emergence of targeted educational interventions to introduce AI concepts to young learners, such as machine learning, natural language processing, and recommendation systems. For example, the AI Chef Trainer activity engages students with the role of data quality and quantity in training machine learning models through an interactive cooking metaphor (Movahed and Martin 2025), while Word2Vec4Kids uses interactive challenges to introduce middle school students to word embeddings (Wiatrek, Verma, and Martin 2025). These efforts illustrate a growing diversity of approaches for making abstract AI concepts tangible and age-appropriate.

Recommendation Systems and Filter Bubbles

Recommendation systems personalize information delivery by ranking and suggesting items based on a user's profile, which can be built from explicit preferences, implicit behavior, or both (McNee, Riedl, and Konstan 2006). While personalization can improve user experience, it can also create "filter bubbles" (Pariser 2011), situations in which exposure to diverse perspectives is reduced. Empirical research has shown that algorithmic personalization can lead to echo chambers in online media consumption (Flaxman, Goel, and Rao 2016). Understanding both the mechanisms and societal consequences of recommendation systems is therefore a critical AI literacy goal, particularly for young learners whose media habits are still forming(Iqbal et al. 2025; Sperling et al. 2025).

Game-Based and Embodied Learning Approaches

Game-based learning has been shown to enhance engagement, motivation, and conceptual understanding across STEM domains (Gee 2003). Embodied learning approaches, in which learners physically or virtually act within a metaphorical representation of a system, can make abstract computational processes more concrete (Schank and Cleary 2014; Abrahamson and Lindgren 2014).

Minecraft has emerged as a particularly versatile platform for educational experiences due to its open-ended environment, accessibility, and active modding community (Tablatin, Casano, and Rodrigo 2023; Karsenti and Bugmann 2019). Its educational branch, Minecraft Education Edition, brings structured lessons to classrooms, fostering collaboration and problem-solving across fields like social studies, computational thinking, and environmental science. Recent educational modules, such as AI Foundations (Minecraft Education 2025a) and Reed Smart: AI Detective (Minecraft Education 2025b), introduce students to basic AI principles including data training, ethical considerations of AI, and pattern recognition. However, these experiences focus on general AI understanding rather than the mechanisms of recommendation systems or the concept of filter bubble formation in these systems.

Figure 1: Initial garden state in BeeTrap-MC. Right: flower attributes. Top: diversity score bar. Bottom: player inventory.

The BeeTrap Approach

The original *BeeTrap* application (Zhou 2024) employed an augmented reality, facilitator-led group activity in which players act as bees, pollinating flowers and observing how repeated similar choices reduce diversity in their environment. This embodied analogy was designed to parallel how recommendation systems shape user experiences and create filter bubbles. Its design combined embodied movement, analogical reasoning, and collaborative discussion, allowing learners to physically experience algorithmic processes such as similarity-based ranking and diversity loss through their actions. The activity was intentionally grounded in image-schema theory, using embodied metaphors such as NEAR–FAR (Hurtienne and Israel 2007) distance to help learners connect algorithmic similarity and diversity to spatial relationships within the augmented environment.

While the original BeeTrap demonstrated promising learning outcomes, it required significant resources, time, and instructor facilitation. The current work, *BeeTrap-MC*, adapts this experience into a self-guided Minecraft-based format intended for shorter, individual play sessions, making it more scalable for broader K–12 AI literacy outreach.

Resource Description and Design

Setup and Mechanics

In BeeTrap-MC, players spawn in a bounded Minecraft garden populated with a variety of flowers that serve as analogies for multidimensional datapoints (see Fig. 1). Each flower is characterized by a set of attributes, which are revealed as textual labels when a player aims at it. A floating beehive at the center of the garden represents the player's evolving "profile." The design builds on the embodied metaphor of the NEAR–FAR image schema (Baur, Wienrich, and Hurtienne 2022) used in the original BeeTrap (Zhou 2024), in which spatial proximity parallels similarity in multidimensional data: flowers that are closer together


Garden 4 – Diversity score: 1408

use RIGHT click to use an item.
scroll (or use numbers) to
change item from inventory

Figure 2: Garden in the fourth pollination round, showing the beehive, connector lines, ranked buds within the pollen circle, and pollens flying toward the hive.


This is the newest garden!

Figure 3: Garden state before and after the first activity, with withered flowers in the distance. Players can revisit rounds using time travel tools.

are more alike, and the beehive's position is updated to the average of all previously pollinated flowers.

The game consists of three activities (corresponding to the learning objectives discussed in the next subsection), with each activity composed of multiple rounds of gameplay. During each round, players receive on-screen prompts, hints, guides, or multiple-choice questions. A toolbar at the bottom of the screen provides access to pollination tools, while the garden diversity score is shown at the top as a progress bar. Each round begins with the player selecting a flower, aiming at it, and using the pollination tool. Each pollination updates the beehive's position to the average of all previously pollinated flowers, effectively visualizing how a recommendation system's user profile shifts based on interaction history. To further reinforce this embodied metaphor, a line is rendered on the ground connecting the beehive to previously pollinated flowers (see Fig. 2). After each pollination, a pollen circle appears around the hive, and a set of flower buds (analogous to recommended items) emerge within this circle. Pollens fly from buds toward the beehive to simulate biological pollination. Buds are then ranked by their Euclidean distance to the hive: the three nearest buds bloom into new flowers, while the three most distant buds wither. The diversity score (calculated as the sum of pairwise distances among all flowers in the garden) is updated accordingly to reflect overall variety.

As players continue pollinating, the garden gradually becomes dominated by similar flowers, illustrating the emergence of a filter bubble. After the first round, players receive two "time travel" tools in their inventory that allow them to revisit previous or subsequent states of the garden (see Fig. 3). In the final activity, players are introduced to strategies for increasing diversity: they gain tools to expand the pollen circle radius (increasing the pool of possible recommendations) and to change the ranking method by toggling a lever in the garden (see Fig. 4), which allows more distant buds rather than only the closest ones to bloom.


Garden 1 – Diversity score: 477

use RIGHT click to use an item.
scroll (or use numbers) to
change item from inventory

Diversifying ranking method enabled!

Figure 4: Diversification Activity, players enlarge the pollen circle and change ranking method to choose the farthest buds to grow and diversify the garden

Learning Objectives and Activities

Following the original BeeTrap design (Zhou 2024), BeeTrap-MC addresses three learning objectives: (1) understanding the filter bubble effect, (2) grasping the inner workings of recommendation systems that cause filter bubbles, and (3) applying diversification strategies to break them.

Activity 1: Experiencing Filter Bubbles. Students begin by pollinating flowers in the Minecraft garden. Repeatedly selecting similar flowers causes the garden to become dominated by that type, while distant flowers wither. The diversity score decreases accordingly, illustrating how filter bubbles form when systems only recommend items aligned with prior choices.

Activity 2: Understanding Inner Workings. The second activity follows the same general flow but provides more in-depth, on-screen explanations. After each pollination the game pauses, prompting players to reflect on what the beehive, flower distances, and pollen circle represent in terms of user profiles and similarity. In later rounds, these multiple-choice questions appear on screen (Fig. 5) to reinforce these concepts and check understanding:

- *What goes into the User Profile(beehive)?*
- *What do the flower buds on the ground represent?*
- *What do numbers above flower buds represent?*
- *What does the pollen circle represent?*
- *How are the flowers located in the garden?*

Each question and its answer choices were intentionally designed to bridge the in-game analogies (e.g., flowers, bee-


What goes into the User Profile(Beehive)?

Figure 5: multi-chioce question from Activity 2

hive, pollen circle) with their algorithmic counterparts in recommendation systems, explicitly prompting players to connect game elements to concepts such as user profiles, item similarity, and ranking processes.

Activity 3: Breaking the Filter Bubble through Diversification. In the final activity, players are introduced to strategies for counteracting homogenization. After pollination they are prompted to enlarge the pollen circle, which reveals more candidate flower buds. They can then change the ranking method by flicking an in-game lever (see Fig. 4), allowing more distant buds to grow instead of only the nearest ones. These mechanics demonstrate how diversification increases the variety of recommended items.

Implementation

Our initial development of BeeTrap-MC began as a *Spigot* plugin, which allowed us to quickly prototype the pollination mechanics inside Minecraft servers. However, this approach limited our control over core game mechanics and data handling. To achieve greater flexibility and tighter integration with Minecraft's rendering and interaction systems, we transitioned to developing a standalone mod using the *Fabric* modding framework over Minecraft Java edition. This choice enabled us to customize both server- and client-side behavior while preserving Minecraft's native feel.

For the dataset, we generate data points similar to the original flower dataset introduced in the BeeTrap application every time. Each flower is characterized by five attributes. In the original implementation, visual features such as petal shape, petal size, leaf size, color, and height were mapped directly onto 3D flower models created in Blender. In BeeTrap-MC, we reimplemented the Classical Metric Multidimensional Scaling (MDS) algorithm to project these five-dimensional datapoints into two-dimensional plane coordinates, preserving the relational structure of the dataset in the Minecraft world. To remain faithful to Minecraft's aesthetic, we employed in-game, low-resolution flower assets. This decision simplified visual fidelity but introduced representational challenges: Minecraft flowers only vary in color. To compensate, we introduced additional abstract attributes—"smell strength," "nectar sweetness," "water needed," and "sunlight needed"—which were displayed as textual data (see Fig. 1). These attributes were less immediately salient than the original visual encodings but allowed us to preserve the multidimensionality of the data within the Minecraft environment.

ID	Gender	Gr.	Race	Prog Exp	MC Exp
P1	Male	9	AA	Low	Mid
P2	Male	8	Hisp./AA	Low	High
P3	Male	9	AA	Mid	Mid
P4	Male	7	AA	Mid	High
P5	Male	8	Lat./AA	Low	High
P6	Female	6	AA	Low	High
P7	Female	7	AA	Low	Low
P8	Male	9	AA	Mid	Mid
P9	Male	7	AA	Low	Mid

Table 1: Participant demographics, grade, prior programming experience, and Minecraft experience (N=9). Programming experience coded as Low = never, Mid = a couple of times. Minecraft experience coded as Low = never or rarely, Mid = a couple of times, High = most weeks or daily.

The Fabric framework enables code injection into Minecraft's Java runtime, letting us modify both server and client logic. On the server side, we implemented game controllers, custom entities, flower mechanics, and networking. On the client side, we handled rendering of entities, custom UI screens, and client-only entity effects.

User Study

Participants

Participants consisted of nine students from a summer camp (grades 6–9) from the same summer camp program as the original BeeTrap paper. Participant demographics are summarized in Table 1. The majority of participants identified as male (7/9) and Black/African American (6/9). Most reported little or no prior programming experience, with only three participants indicating mid-level experience ("a couple of times"). Minecraft familiarity was generally higher, with most participants reporting mid to high experience, meaning the choice of platform was accessible for this audience.

Procedure

Each participant completed the BeeTrap-MC activity individually on a laptop, with sessions lasting approximately 45 minutes. The procedure consisted of three phases:

1. **Pre-survey and knowledge test:** Participants completed a short demographic questionnaire and a pre-test with seven open-ended questions assessing their understanding of recommendation systems, filter bubbles, and related AI concepts.
2. **Gameplay session:** Participants played through the three BeeTrap-MC activities described in Section 3, each lasting approximately 10 minutes. Prompts and multiple-choice questions were embedded in-game to encourage reflection during play. No facilitators provided hints; the experience was designed to be self-guided.
3. **Post-survey and interview:** Participants repeated the same seven-question knowledge test and completed a

short usability survey (Brooke 2013). Semi-structured interviews followed, focusing on participants' reflections on the game, their learning, and their perceptions of AI systems.

Data Collection

We collected four sources of data during the study:

1. **Pre- and post-tests:** Participants completed identical seven-item open-ended assessments before and after gameplay. These questions were adapted directly from the original BeeTrap study (Zhou 2024) to allow comparability. The items were:
   - Q1. What AI-based recommendation system have you used?
   - Q2. What is a filter bubble? What are the negative impacts of it?
   - Q3. What does a user profile include?
   - Q4. Which object is more similar to (2,4,2,4,2)?
     a) (2,2,4,2,2) b) (4,4,2,4,4)
   - Q5. How do recommendation systems predict what you like?
   - Q6. What happens if recommendations are only based on your history?
   - Q7. Order the steps of a recommendation system:
     - Update profile
     - Select item
     - Recommend similar items
     - Rank items

   These questions address the first and second learning objectives, since diversification wasn't improved significantly in the original paper and this was a mere replica in minecraft we didn't assess that but we report on that in the qualitative results section.

   We graded the open-ended responses using the same 0–3 rubric and conceptual categories defined in the original study. The rubric evaluated the conceptual understanding demonstrated in each response (0 = no understanding, 1 = partial/incomplete, 2 = mostly correct, 3 = fully correct explanation with reasoning). For example, for Q2, a score of 3 was assigned when students mentioned that (1) only similar or limited types of information or items are recommended, (2) these reinforce prior preferences or selections, and (3) the outcome reduces diversity or leads to negative effects such as the "flowers dying." Partial responses mentioning only one or two of these key ideas received a score of 1 or 2, while irrelevant or incorrect answers received 0.

   Two researchers independently scored all pre- and post-test responses following this rubric. Inter-rater reliability was assessed using Cohen's κ, which yielded a value of $\kappa = 0.82$, indicating strong agreement between raters. Any discrepancies were discussed and resolved through consensus before the final analysis.
2. **Interviews:** Semi-structured interviews were conducted after gameplay. Students were asked to explain or elaborate on their pre- and post-test answers, providing insight into their reasoning and how their understanding may have shifted.
3. **In-game data:** Screen recordings documented gameplay behaviors and on-screen interactions. The system also generated logs capturing changes in garden diversity scores, start and end of pollination rounds, and item ranking events.
4. **Embedded questions:** Multiple-choice questions were integrated into gameplay (Activity 2) to probe participants' understanding of recommendation concepts in real time.

Q	Pre M(SD)	Post M(SD)	Diff M(SD)	p
Q1	1.61 (1.11)	2.50 (1.00)	0.89 (0.96)	**.031**
Q2	0.06 (0.17)	1.72 (1.20)	1.67 (1.22)	**.016**
Q3	0.22 (0.44)	1.33 (0.90)	1.11 (1.08)	**.039**
Q4	0.33 (1.00)	0.67 (1.32)	0.33 (1.00)	1.000
Q5	0.61 (0.65)	1.83 (1.22)	1.22 (1.39)	**.047**
Q6	0.22 (0.44)	1.50 (1.27)	1.28 (1.15)	**.031**
Q7	0.28 (0.67)	1.44 (1.07)	1.17 (1.03)	**.016**
	0.48 (0.64)	1.57 (1.14)	1.10 (1.12)	**.001**

Table 2: Pre- and post-test scores on 7 conceptual questions (0–3 scale) with the overall scores in the bottom. Significant $p < .05$ values in bold.

Analysis and Results

Learning Gains

To assess learning gains, we compared participants' performance on pre- and post-assessments. Across all seven questions, the mean pre-test score was $M = 0.15$ ($SD = 0.43$), while the mean post-test score increased to $M = 0.52$ ($SD = 0.34$). Although this reflects a clear improvement, the average post-test score still indicates that participants achieved only partial mastery of the target concepts, suggesting room for deeper understanding.

Because the data were not normally distributed (Shapiro–Wilk test: $W = 0.91, p < .001$), we used a non-parametric test for paired samples. A Wilcoxon signed-rank test showed that post-test scores were significantly higher than pre-test scores, $W = 28.5, p < .001$.

Table 2 summarizes mean scores, standard deviations, mean differences, and corresponding p-values for each question. Significant learning gains were observed on most questions, particularly those targeting understanding the concept of filter bubble, being aware of it and recommendation system mechanics.

System Usability

We measured usability with the *System Usability Scale* (SUS) (Brooke 2013). Scores range from 0 to 100, with 68 as the benchmark for average usability. BeeTrap-MC received a mean SUS score of $M = 63.33$ ($SD = 27.50$), slightly below the benchmark, with high variance indicating mixed perceptions of ease of use.

Interview responses provide further insight into these mixed ratings. Several participants noted that, although

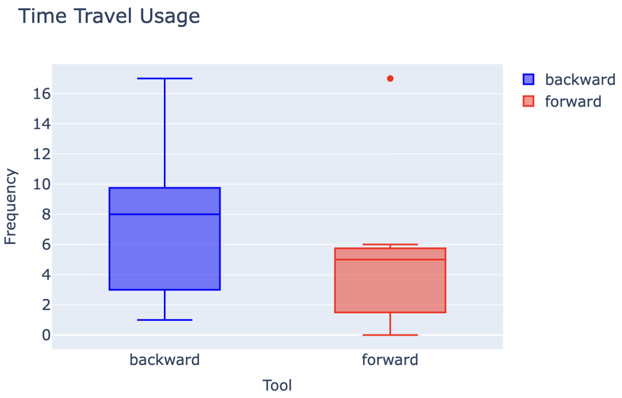

Figure 6: Box Plot for backward and forward time travel

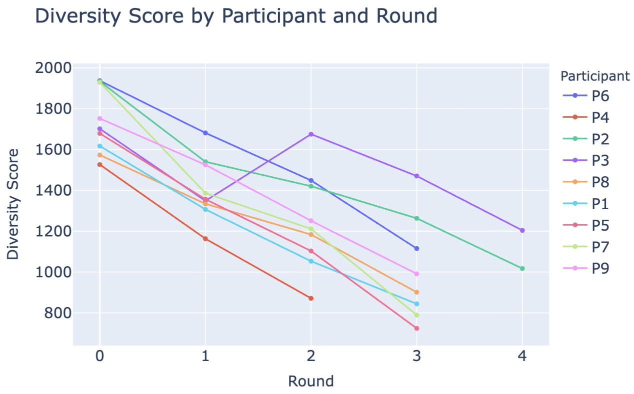

Figure 7: Flower Diversity in Activity 1 per participant over rounds

BeeTrap-MC was presented as a game, the frequent pauses for concept explanations and reading text felt cumbersome, with one student describing the experience as “boring and repetitive.” Others remarked that while the activity took place in Minecraft, its mechanics felt very different from their prior experiences with the game. This suggests that participants’ high pre-engagement with Minecraft may have set expectations that an educational game built in the same environment could not fully meet.

Gameplay Patterns

In addition to pre-/post-assessments, we analyzed in-game behavioral data to better understand how participants interacted with BeeTrap-MC.

Use of Time Travel Tools. BeeTrap-MC included “backward” and “forward” time travel tools that allowed players to revisit earlier or later states of the garden after a round of pollination. Our logs indicate that most participants used these tools at least a few times, with varying frequency. Notably, Participant 8 (P8), who achieved the highest post-test score (0.76), made extensive use of this feature, employing both backward and forward travel 17 times each. This suggests that iterating over the garden states may have supported deeper reflection and learning. This finding aligns with the original BeeTrap study, where iteration and comparison across garden states was also identified as a meaningful learning mechanism.

Garden Diversity Over Time. We also tracked the diversity score of each participant’s garden, measured as the sum of pairwise distances among flowers. In every case, diversity decreased as the game progressed through the first activity, indicating that all participants successfully experienced the “filter bubble” effect. This outcome mirrors the original BeeTrap results, confirming that the core metaphor of narrowing diversity was preserved in the Minecraft adaptation.

Figure 6 illustrates usage patterns of the time travel tools, while Figure 7 shows the garden diversity trajectories per participant.

Limitations, Discussion, and Future Work

Our study of BeeTrap-MC demonstrates both the potential and the limitations of adapting an embodied AR activity into a self-guided Minecraft environment for AI literacy education. The only question that did not show measurable learning gains was the one concerning multidimensional data similarity (Q4). We attribute this to the reduced representational fidelity of Minecraft flowers compared to the original AR implementation, where multiple visual attributes (petal shape, size, leaf size, etc.) could be mapped directly. In Minecraft, only color could be varied natively, and while we added abstract textual attributes (e.g., “smell strength,” “nectar sweetness”), these were less salient and less likely to be internalized.

Compared to the original BeeTrap, BeeTrap-MC used substantially less time (30 minutes vs. two hours) and required students to work individually rather than in facilitated groups. This difference influenced interaction patterns: in the AR workshop, students often discussed hints and multiple-choice questions collaboratively, whereas in the Minecraft version, players tended to skip over prompts. Even those who later answered correctly on the post-test sometimes failed to engage with the in-game multiple-choice questions. This suggests that for individual digital formats, future work might explore incorporating pedagogical agents or conversational guides rather than relying solely on on-screen text prompts.

Another difference emerged in terms of embodiment and movement. In the AR version, participants could physically walk around the room, but in practice they did not move very extensively. In Minecraft, despite the limited embodiment, the players navigated the garden much more actively and fluidly, as evidenced in the screen recordings. This suggests that the digital environment encouraged faster exploration and potentially heightened engagement, even if the embodied fidelity was reduced. At the same time, because Minecraft is already a highly engaging platform for youth, future iterations of BeeTrap-MC—and educational games in Minecraft more generally—should incorporate more of the native mechanics and “feel” of Minecraft to avoid a sense of dissonance between expectations and gameplay.

Due to the small sample size, no statistically significant patterns were observed between participants' gender, prior programming experience, or previous Minecraft experience. Scaling up the study will be necessary to investigate whether such factors influence learning outcomes. Additionally, participants' prior familiarity with Minecraft and potential novelty effects may have influenced engagement and learning outcomes, particularly in the absence of a control group or direct comparison to the original BeeTrap or a non-game baseline.

The relatively low SUS score ($M = 63.33$) and feedback about pacing suggest that future iterations should streamline instructional pauses and better align educational mechanics with native Minecraft gameplay to maintain engagement while conveying conceptual depth.

Finally, in terms of development, creating BeeTrap-MC presented technical challenges. Although the Fabric modding framework provided the necessary flexibility, we encountered difficulties due to limited and fragmented online documentation. Our future work is two-fold. First, we plan to deploy an online browser-based version of BeeTrap-MC with automatic logging, allowing us to reach a wider audience and collect more comprehensive gameplay data. Second, we aim to investigate integrating the BeeTrap experience into Minecraft Education to explore its classroom scalability and alignment with formal curricula. Despite these challenges, our implementation shows that Minecraft is a promising platform for AI literacy outreach, providing accessibility and scalability while preserving the core embodied analogy of the original BeeTrap.

Conclusion

BeeTrap-MC was developed with the goal of making AI literacy, particularly the concept of filter bubbles, more accessible to students through a widely available platform. Our evaluation showed that the Minecraft-based adaptation was effective in supporting measurable learning gains and helped students recognize both the mechanics and consequences of recommendation systems. At the same time, the study highlighted important challenges in adapting embodied educational activities into more accessible formats. Differences between the original AR-based, group-facilitated BeeTrap and the self-guided digital BeeTrap-MC underscore the trade-offs of individual versus collaborative play, as well as high-fidelity embodied interaction versus simplified digital metaphors. Taken together, our findings suggest that while Minecraft offers a promising and scalable medium for AI literacy outreach, careful design attention is needed to preserve key aspects of engagement, collaboration, and embodied learning in translation to digital platforms.

Acknowledgments

This work is supported by the National Science Foundation (NSF) RETTL program (Award No. 2238675). We would like to thank Jeremy Smith, Kelvin Sanchez-Orona, and the teaching assistants at the Freedom Scholars Learning Center for their generous support, as well as children who participated in our study and their parents.

References

Abrahamson, D.; and Lindgren, R. 2014. Embodiment and Embodied Design. In Sawyer, R. K., ed., *The Cambridge Handbook of the Learning Sciences*, 358–376. Cambridge: Cambridge University Press.

Baur, C.; Wienrich, C.; and Hurtienne, J. 2022. Form follows mental models: Finding instantiations of image schemas using a design research approach. In *Designing Interactive Systems Conference*. New York, NY, USA: ACM.

Brooke, J. 2013. SUS: a retrospective. *Journal of usability studies*, 8(2).

Bulger, M.; and Davison, P. 2018. The promises, challenges, and futures of media literacy. *J. Media Lit. Educ.*, 10(1): 1–21.

Casal-Otero, L.; Catala, A.; Fernández-Morante, C.; Taboada, M.; Cebreiro, B.; and Barro, S. 2023. AI literacy in K-12: a systematic literature review. *Int. J. STEM Educ.*, 10(1).

Flaxman, S.; Goel, S.; and Rao, J. M. 2016. Filter bubbles, echo chambers, and online news consumption. *Public Opin. Q.*, 80(S1): 298–320.

Gee, J. P. 2003. *What video games have to teach us about learning and literacy*. Macmillan.

Hurtienne, J.; and Israel, J. H. 2007. Image schemas and their metaphorical extensions: intuitive patterns for tangible interaction. In *Proceedings of the 1st International Conference on Tangible and Embedded Interaction*, TEI '07, 127–134. New York, NY, USA: Association for Computing Machinery. ISBN 9781595936196.

Iqbal, M.; Singh, K.; Khan, S.; Osho, O.; Sidnam-Mauch, E.; Bannister, N.; Caine, K.; and Knijnenburg, B. 2025. Teaching AI awareness in middle school classrooms: Design, implementation and evaluation of two education modules on algorithmic bias and filter bubbles. *Computers and Education: Artificial Intelligence*, 8: 100425.

Karsenti, T.; and Bugmann, J. 2019. Can Minecraft improve student engagement? A qualitative study of student experiences in secondary school. *Education and Information Technologies*, 24: 2341–2355.

McNee, S. M.; Riedl, J.; and Konstan, J. A. 2006. Being accurate is not enough: How accuracy metrics have hurt recommender systems. In *CHI'06 extended abstracts on Human factors in computing systems*, 1097–1101. ACM.

Minecraft Education. 2025a. AI Foundations. https://education.minecraft.net/en-us/discover/ai. Accessed: 2025-11-27.

Minecraft Education. 2025b. Reed Smart: AI Detective. https://education.minecraft.net/en-us/lessons/reed-smart-ai-detective. Accessed: 2025-11-27.

Movahed, S. V.; and Martin, F. 2025. AI Chef Trainer: Introducing Students to the Importance of Data in Machine Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(28): 29220–29227. EAAI Symposium: Resources for Teaching AI in K-12.

Pariser, E. 2011. *The filter bubble: What the Internet is hiding from you*. Penguin UK.

Schank, R. C.; and Cleary, C. 2014. *Learning by doing: A comprehensive guide to simulations, computer games, and pedagogy in e-learning and other educational experiences*. John Wiley & Sons.

Short, D. 2012. Minecraft as a teaching tool: One case study. *British Journal of Educational Technology*, 43(4): E6–E8.

Sperling, K.; Stenliden, L.; Mannila, L.; Hallström, J.; Nordlöf, C.; and Heintz, F. 2025. Perspectives on AI literacy in middle school classrooms: An integrative review. *Postdigit. Sci. Educ.*

Tablatin, C. L. S.; Casano, J. D. L.; and Rodrigo, M. M. T. 2023. Using Minecraft to cultivate student interest in STEM. *Front. Educ.*, 8(1127984).

Touretzky, D.; Gardner-McCune, C.; and Seehorn, D. 2022. Machine learning and the five big ideas in AI. *Int. J. Artif. Intell. Educ.*

Wiatrek, N.; Verma, Y.; and Martin, F. 2025. Word2Vec4Kids: Interactive Challenges to Introduce Middle School Students to Word Embeddings. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(28): 29228–29235. EAAI Symposium: Resources for Teaching AI in K-12.

Zhou. 2024. Bee and I Need Diversity! Breaking Filter Bubbles through Embodied Learning. In *Proceedings of the 19th International Conference on the Foundations of Digital Games*. ACM.