From Tool to Partner: Exploring the Roles of Embodiment on AI Agent in Pair Programming

Xiaoran Yang, Yang Zhan, Noboru Matsuda, Qiao Jin
Dept. of Computer Science, North Carolina State University, Raleigh, United States
xyang49@ncsu.edu, yzhan3@ncsu.edu, nmazda@gmail.com, qjin4@ncsu.edu

***Abstract*—Pair programming with AI often faces challenges in productive communication and engagement. Integrating embodiment offers a promising solution by making AI a more engaging and context-aware programming partner. To explore how embodied AI agent supports programming learning and affects user experiences, we designed a virtual reality (VR) programming environment with Wizard-of-Oz-controlled AI agents. Our study collected data from 18 participants through knowledge acquisition assessments and interviews. The results showed that embodiment improved engagement, enhanced communication efficiency, and offered emotional support. Specifically, the incorporation of embodied actions allows users to perceive the AI agent as a "programming partner" and introduces many interactions that resemble those shared with real-life partners. However, the effectiveness of embodied actions in supporting users with programming tasks depends on the timing and accuracy of those actions. This study reveals the potential of embodied AI agents in advancing programming education and provides valuable design insights for creating more intuitive and supportive AI programming partners.**

***Index Terms*—Pair Programming, Virtual Reality, Human-AI Collaboration, Embodied AI Agents**

I. Introduction

Programming is increasingly recognized as a critical skill, providing the foundation for developing problem-solving and computational thinking skills essential in various domains [1]–[3]. While programming offers numerous benefits to students, learning programming skills can be challenging due to the abstract nature of knowledge structures, the practical demands of implementation [4]–[6], and the lack of personalized programming experiences [7].

With the advent of Large Language Models (LLMs), programming with Artificial Intelligence (AI), exemplified by Github's Copilot [8], introduces a new paradigm of personalized programming education to respond to this challenge. Similarly to human-human pair programming, the user and AI agent work on the same programming task, where the user takes responsibility for program implementation while AI provides complementary support in program planning, synthesis, and revision (termed 'pAIr programming') [9]. For novice programmers, pAIr programming has shown significant benefits, including improved task performance, reduced programming time, and lower levels of anxiety and discouragement [10], [11]. Through the integration of chat and code generation, those AI systems in programming transcend mere tool functionality, taking on roles analogous to instructors or collaborative partners.

However, some researchers criticize the description of pAIr programming as human-human pair programming [9], arguing that the analogy is superficial because it lacks key elements of human-human pair programming, such as productive communication. This unsatisfied communication has a negative effect on learning efficacy. Additionally, the lower solution quality compared to human-human pair brings possible buggy programs to novices [12], which can raise more need for explanation. Prior research [13] suggests that embodied interaction, which uses the physical-world couplings between actions and their effects through analogies and metaphors, offers a promising approach to making communication more intuitive and engaging, and potentially more productive. In education, embodied interaction has been explored as a method to contextualize programming knowledge, using physical actions and spatial references to improve expression and facilitate communication [14]–[16]. For example, body movements and gestures can contribute to stronger social engagement [17] and support the use of metaphors to represent concepts [18].

Inspired by these merits, it is worth exploring how the embodiment of the AI agent impacts pAIr programming. In this paper, we define an embodied AI agent as a virtual entity that can recognize and generate both verbal and non-verbal cues (e.g., gestures and body animations) and provide context-aware support (e.g., task progress) [19]–[21]. Unlike traditional embodied conversational agents (ECAs), embodied AI agents leverage broader open-world knowledge to handle multimodal inputs, which allows for more efficient communication and deeper contextual understanding [22]. We created a block-based programming environment in VR using a Wizard-of-Oz setup with a virtual programming partner—an embodied AI agent that can gesture, move, and sense user actions and scene changes. To understand how embodiment (e.g., gestures and movements) influences the pAIr programming process, we evaluated the system with 18 participants. Specifically, we aim to answer the following research questions:

- **(RQ1)** How does the embodiment of an AI agent affect learners' programming knowledge acquisition?
- **(RQ2)** How does the embodiment of an AI agent affect user programming experiences?

II. Related Work

A. Human-AI Pair Programming

Pair programming, where two programmers collaborate on the same task, has been shown to improve productivity and code quality, as well as facilitate knowledge transfer [23]–[25]. However, traditional pair programming faces several challenges, including cost-efficiency, scheduling conflicts, and personality mismatches. Its effectiveness is also influenced by factors such as task complexity and the relative expertise of programmers [25], [26]. pAIr programming, where an AI agent replaces a human partner, addresses these challenges by offering comparable outcomes in productivity, code quality, and self-efficacy [27]. AI agents adapt to varying expertise, helping mitigate mismatched expertise, and tools like PairBuddy [28], [29] support both technical and soft skills, fostering knowledge transfer. Additionally, pAIr programming alleviates logistical issues, such as scheduling conflicts and partner compatibility [30]. Despite these advancements, AI agents currently fall short in areas such as engaging in rich and productive discussions, which are hallmarks of human-to-human collaboration in pair programming [27]. Prior study has emphasized the value of embodied interactions in pair programming, such as shared gaze awareness and gestures, as they enable concurrent code viewing and reduce the need for explicit references, thereby enhancing collaboration [31]. Therefore, exploring the role of embodiment in pAIr programming is important. Many ECAs equipped with visual appearance and embodied actions have been used in educational settings [32]–[34]. However, most existing ECAs provide only limited context-aware support, typically rely on scripted interactions and lack proactive sensing [35]. As a result, they may miss opportunities for more dynamic and adaptive support. To address this, we focus on embodied AI agents with context-aware capabilities and examine their role as programming partners in pAIr programming, an area that has received limited attention in prior research.

B. Programming in VR

VR has shown promise in programming education by improving conceptual understanding and motivation through immersive, low-cost, and flexible 3D environments [36]–[41], while also helping reduce health risks from prolonged sitting [42]. Systems like MR-FTC, Hack.VR, Cubely, and BlocklyVR support code visualization and hands-on learning, especially for beginners [42]–[45]. However, existing research largely focuses on individual learning, with limited exploration of pair programming in VR, especially with AI agents. Prior work on virtual avatars in traditional pair programming highlights the potential for enhanced collaboration [31]. Building on this, our study designs a VR-based system where users program alongside an embodied AI agent. VR enables embodied interaction, spatial awareness, and social presence, making it a powerful medium for investigating how embodiment enhances the AI agent's role in programming.

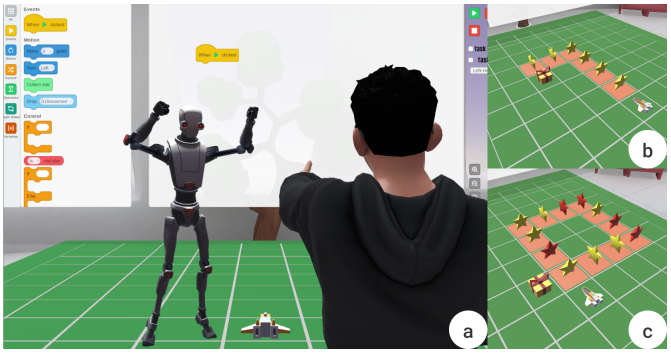

Fig. 1. Overview of the embodied AI programming environment. (a) The AI agent interacts with participants either proactively based on their progress or reactively to their inquiries. (b, c) Map sandboxes for the structured programming tasks: (b) Task 1 requires using a loop structure to collect all stars and a gift. (c) Task 2 requires using loop, condition, and variable constructs to collect three red stars and a gift.

III. Methods

To investigate the role of embodiment for the AI agent in programming learning, we employed a hybrid Wizard-of-Oz (WoZ) design [46]. Participants were let to believe they were interacting with an autonomous embodied AI agent, but actions were partially controlled by researchers. This approach enabled us to explore the impact of embodiment in a controlled environment, even without fully implemented system functionality. Our study included two task types using embodied AI agent: structured tasks and unstructured tasks (see Section III-B2). For structured tasks, we used a within-subjects design (see Section III-C) with two conditions: embodied AI agent and voice-only AI agent.

A. Participants and Setting

Participants were recruited through email outreach at the university. Eighteen participants (12 females, 6 males), aged from 19 to 27 years (M = 24, SD = 2.35), participated in this study. Among them, 4 participants were frequent VR users, 11 had used it fewer than three times and 3 never used VR. Regarding programming experience, 12 participants reported no prior experience, 2 had only basic experience but self-identified as novices, and the remaining 4 had more advanced experience with programming concepts and tools. Participation in this study was voluntary and no compensation was provided.

B. pAIr Programming Systems

1) Programming Environment: We developed the VR programming environment using Unity and Oculus Quest 3. It includes: (1) **Block-based programming panel**: We adopted Blockly's block-based programming approach [47], creating a panel where users construct programs by blocks. Users select code blocks with a laser pointer and drag them using the right trigger button. Compatible blocks snap together automatically. Pressing the left rear button opens a menu to delete selected blocks. The panel also supports quick deletion, zooming, and execution like other block-based editors. (2) **Map sandbox**: To provide an intuitive visualization of the program execution, we

incorporated a map sandbox to display real-time programming results. As shown in Fig. 1 (b, c), users can arrange programming blocks to control the airplane's movement in the sandbox and collect stars.

2) Task Design: In order to investigate pAIr programming with different goals, we designed two types of programming tasks: **(1) Structured tasks**: predefined programming problems (Tasks 1 and 2). Fig. 1b shows Task 1 that requires participants to use programming concepts of loops to control an airplane to collect all stars and a gift along a specified path. Task 2 (Fig. 1c) involves using programming modules related to loops, conditions, variables to navigate the airplane along a specified path to collect three red stars and eventually a gift. **(2) Unstructured task:** an open-ended design challenge with no set goals. Completion is based on participants' subjective understanding, particularly their sense of aesthetics and harmony. Specifically, participants need to control an airplane to drop objects and decorate the entire map, They can use all the programming modules to complete this task.

3) Embodied AI Agent: AI agent can dynamically move within the space and interact with users by giving both reactive responses and proactive responses through spatial audio and embodied actions to assist programming (Fig. 1a). **Reactive responses** occur when users ask the AI agent a question. While the AI agent provides voice feedback converted from LLM-generated text, the researcher uses action placeholders in the text to control the AI agent to perform corresponding actions. **Proactive responses** controlled by researchers occur and were triggered based on observed hesitation or task-related pauses. These responses enable context-aware support, where the system not only reacts to task events (e.g., task completion) but also responds to user states (e.g., session greetings, periods of idleness). This context-aware support also uses user-captured screenshots of their surrounding environment to guide appropriate responses.

We implemented an embodied AI agent (GPT-4o) with a human-like appearance, as research shows such agents promote intuitive interaction in XR [21]. From a Unity animation asset (Dialogue Anims[1]), we selected 12 communication-focused actions, including hand gestures and body movements like pointing at a block. These cover gesture types [48], [49]: deictic (e.g., pointing to guide attention), emblematic (e.g., thumbs-up for praise), and metaphoric (e.g., thinking to show abstraction). Those are commonly used in everyday communication and have been shown to support learning [50]. The LLM prompt specifies the AI's role, task description, learning goal, and output format (e.g., "As a supportive and engaging AI programming partner, you guide users to complete tasks, using dynamic actions and concise explanations to enhance learning, motivation, and understanding while providing emotional support and tailored feedback."). The prompt output format consists of response text accompanied with action placeholders (e.g., "[Thumbs-up] Good job!"); in some cases, actions may also appear independently (e.g., "[Celebrate]" when user reached the milestone) [2].

4) Voice-only AI agent: To give participants a clear comparison and better isolate the impact of embodiment, we included a voice-only AI agent that used the exact same prompts as the embodied agent but without the visual representation. The verbal content remained consistent across two agents.

5) Wizard-of-Oz Controls: In this hybrid WoZ setup, reactive response generation and action selection are automated by the LLM model, while researchers manually control the playback of actions, adjustments to the AI agent's position, and the trigger of proactive responses. Users can interact with the AI agent via audio input and utilize the controller to take a screenshot in VR. Researcher A handles the real-time screenshot and the execution of the GPT script. Upon receiving inputs, the LLM analyzes the user's context and generates a text response embedded with action placeholders. Researcher B monitors Researcher A's screen to identify action placeholders and uses the keyboard to control the embodied actions. To enhance the LLM's understanding capabilities, we predefine standard answer images for the three tasks. The LLM compares the user-uploaded images with these standard images to better understand the user's current programming state and provide targeted feedback.

C. Study Procedure

The study lasts around 90 minutes. First, participants signed an informed consent form, finished a demographic questionnaire, and received a guided tutorial in experiencing the system. They then completed two structured tasks, with the order of agent conditions counterbalanced, and one unstructured task only with embodied AI agent in a fixed sequence. Participants completed a knowledge acquisition assessment before and after (pre/post-test) each structured task. Both the pre-test and post-test contained one question, requiring participants to observe a preset block program and select the correct execution result along with explaining its justification from provided options. This design aimed to prevent credit for correct answers based solely on guessing. Only when both the choice and the explanation are correct, it is counted as correct (100%). Other cases are counted as incorrect (0%). The knowledge acquisition assessment were developed by researchers and programming instructors to ensure accurate measurement of learning outcomes. The assessment questions were specific to the content of each task.

Finally, we conducted a semi-structured interview lasting 15 minutes. The interview focused on general perceptions of AI agents and its embodiment and future designs. Specifically, we emphasized the evaluations of the AI agent's embodied actions, their views on the performance of the AI agent's embodiment in two different task types, and their opinions on which type of task benefited more from embodied actions.

[1]https://assetstore.unity.com/packages/3d/animations/dialogue-anims-222285

[2]Full details of prompts and embodied AI agent action lists are available at: https://github.com/IMYXR/VLHCC2025

IV. Results

A. Knowledge Acquisition

To answer RQ1, knowledge acquisition for each condition for structured tasks was calculated as the difference between the post-test and the pre-test score. The paired-sample Wilcoxon signed-rank tests [51] confirmed that both agents can support programming learning: average knowledge assessment score increased from 55.5% (SD = 0.496) to 88.9% (SD = 0.314) in the voice-only AI condition and from 50% (SD = 0.5) to 88.9% (SD = 0.314) in the embodied AI condition. However, a comparison of knowledge acquisition between the two conditions revealed no significant difference in the magnitude of improvement ($W = 9.0$, $p = 0.739$).

B. pAIr Programming Experiences

We conducted a thematic analysis of our interview data [52] to answer RQ2. Three authors (all trained in the Glaser method [53]) independently coded the transcripts using open coding. Discrepancies were resolved through iterative discussions and consensus, with constant comparison ensuring consistency. Themes were then iteratively refined to ensure they were grounded in the data and reflected key patterns across the dataset.

1) Embodiment improves understanding and engagement for programming (T1): Many participants described that the embodied interaction could intuitively point out key and problematic programming blocks, which improved clarity when the AI agent answered their questions. As P12 highlighted the importance of the pointing gesture *"it could identify where in your code there might be issues, point to those locations, and accompany this with its explanations." (P12)* Moreover, P16 valued the agent's nodding when requested a program check, since it provided more intuitive feedback compared to the redundant voice response. The embodiment also increases motivation by encouraging actions such as clapping, nodding, or celebrating, making the programming experience more engaging and supportive. *"I think the thumbing up, and this, celebrating, had a greater impact on me since they gave me support and let me be confident." (P3, P16)* It is worth noting that the embodied actions of the AI agent also brought emotional encouragement, providing the feeling of companionship (P5, P6, P9) and the sense of security (P6) even in the absence of voice commands (e.g., *"cheer gestures can express emotional support without speech" (P16, P18)*).

2) Embodiment influences user's trust on the agent (T2): By serving as a confirming and emphasizing cue that signals the importance of the interaction, it makes the response more persuasive, encourages users to trust its accuracy, thereby influencing their decision-making process and programming outcomes. P1 described the embodied agent as more expressive and human-like compared to a rigid, computer-like system, which made it easier to trust and more likely to accept its suggestions. *"Such trust makes me more willing to interact with it and increases the positivity of the interaction." (P1)* However, when actions feel out of place or unnatural, they can undermine trust. Some participants questioned the agent's reliability in such moments: *"It makes a thumbs-up gesture even when I haven't done anything, which feels really strange to me" (P1)*; *"it keeps repeating the same motion-it's kind of like a twitch... waving its hand—doing it once or twice would be enough." (P17)*

3) Embodiment may introduce distraction during focused problem-solving moments (T3): Participants noted that inappropriately timed embodied interactions can distract them and disrupt their concentration. For example, both P4 and P7 noted that the embodied interaction (especially proactive actions even without accompanying speech) should not appear when they *"paid attention to their interested objects" (P9)*. P4 noted that he forgot what AI said because the embodied actions diverted his attention from the voice. In contrast, many participants preferred reactive embodied responses (i.e., actions triggered by explicit user input), as the presence of such actions was mentally anticipated, reducing the sense of disruption and allowing them to better focus on the task. P10 noted, in reactive response, that the embodied interaction did not disrupt her attention as it just *"quietly support and assist you"* when needed. P1 also noted that embodied interactions felt more appropriate in reactive responses as *"passive interactions make users feel a stronger sense of control over the task"*, thus they can focus on the task.

4) Task types influence desired embodied behaviors (T4): Participants expressed different preferences for embodied actions according to the task type. In structured tasks, the programming-related embodied actions were valued for its practical utility in directly supporting the problem solving for task completion and saving time. For example, P5 preferred the embodiment in the structured task because it offered more timely instructional feedback compared to the open-ended design task, where, as explained, *"there was no right or wrong regardless of what I did." (P5)* Similarly, two participants (P4, P12) reported that they knew their programs were wrong immediately when the embodied AI agent acted *"rejecting"*, which was faster than having to listen to the entire audio before realizing there was a mistake. It is worth noting that participants' expectations for the embodiment of AI agent shifted in unstructured tasks, focusing less on programming-related suggestions (e.g., pointing) and more on emotional and design support, such as providing companionship and encouragement (P3, P9, P16). P4 and P5 attributed this shift to the unstructured task being perceived as more relaxing and allowing for freer exploration. The lower sense of pressure and performance demands in the unstructured task created more opportunities for experiencing emotional support, as explained by P16 *"I focused more on problem solving in the structure tasks, . . . , the unstructured task was like a game where the AI agent became my partner"*.

V. Discussion

We examined how embodiment of the AI agent supports programming learning (RQ1) and affects user experience (RQ2) through a WoZ study. For **RQ1**, learners showed

knowledge gains in both conditions but the improvements were not significantly different between the conditions. For **RQ2**, qualitative results showed that embodiment supported programming by improving clarity and motivation (T1), shaping trust that influenced users' coding decisions (T2), occasionally distracting during focused tasks (T3), and serving different roles across task types—enhancing efficiency in structured tasks and offering emotional support in open-ended ones (T4).

A. Design Implications

1) Matching embodiment with users' programming status: Although our results did not show significantly higher knowledge acquisition with embodied AI agent compared to the voice-only agent, findings still indicate that embodied representation and actions can enhance learners' motivation and engagement (T1), which aligns with prior ECA research in education [32]–[34]. However, as noted in T2 and T3, poorly timed or unnatural embodied actions could be distracting and reduce the trust of AI agent. Trust plays a critical role in human-AI collaboration during decision-making [54], [55], ultimately impacting programming outcomes. We found that poorly timed embodied actions, especially unexpected proactive gestures, can distract users from both the task and the AI's verbal content. Previous work has investigated the detection of appropriate moments for proactive interaction [56]. Tang et al. [57] proposed using LLMs to define event schema for effectively summarizing, narrowing down, and tracking subtle variations in student behavior. Therefore, to design a more intelligent embodied programming partner while reducing distractions, future work should consider using learning analytics approaches to infer the right moment for embodied actions, by analyzing students' step-by-step actions to detect states such as struggle or system misuse [58]. In VR environments, additional cues like eye gaze and head orientation can help estimate user attention [21], allowing for more precisely timed and contextually appropriate embodied support during programming tasks. These cues can also be extended to other XR platforms and integrated with traditional programming environments to enhance contextual information.

2) Exploring customized embodied representations for programming: Previous research has pointed out that avatar embodied representation influences user emotions [59] and trust [60], which is also echoed in our results (T1 and T2). To further strengthen users' engagement and trust in AI in pAIr programming, future systems could consider personalize the embodied agent's appearance and embodied expression by leveraging the memory capabilities of LLMs [61] and adapting to users' linguistic and behavioral characteristics [62]. The personalized agent can be further deployed in non-XR programming spaces as a 2D partner for standard IDEs. Additionally, T1 highlighted that embodied actions improves the clarity of the AI agent's responses. Future research should further improve multimodal expressiveness and explore personalizing embodied behaviors based on the cultural context and background of users to ensure clarity. For example, speaker gesture styles can influence language comprehension [63] and the meaning of gestures can vary between different cultures, which can cause confusion in communication [64]. Considering that overt embodied behaviors may distract users during periods of focused attention (T3), future research could explore more subtle alternatives, such as facial expressions [65], or reduce the visual presence of the embodied AI by minimizing its size or increasing its transparency to mitigate distractions during these critical moments. These strategies could be further refined through user-customized visual designs, such as adjusting visibility, expression intensity, or animation frequency to better match individual preferences and attention states.

3) Aligning embodiment with type of programming task: T4 indicated that the design of embodiment should match participants' expectation at different types of programming task, which further influenced interaction willingness and perceived usefulness. Hoffmann et al. [66] was found that the presentation of embodied actions differs in their application between goal-oriented and persuasion-oriented scenarios, with distinctions in competence, entertainment value, and informational content. For quiz-like tasks, such as the structured tasks in our study, users are assigned clearer goals, and thus may express more willingness to receive feedback of the programming content from quick embodied reactions. This suggests the value of instructional embodied actions in structured tasks, which may not require accompanying verbal explanations. Such nonverbal cues can help users quickly identify correctness and guide their reasoning processes without directly providing answers. In unstructured or exploratory programming tasks, users are more inclined toward free exploration and brainstorming [67], which reduces the need for direct guidance on specific answers. For such tasks, emotional support should be emphasized to enhance users' motivation for continuous exploration. Future research should design embodied actions based on specific learning objectives.

B. Limitations and Future Work

The small sample size and the limited number of programming tasks and embodied actions used in this study may restrict the generalizability of our findings. Future work could broaden the task scope and vary embodiment designs (e.g., facial expressions, expanded gesture sets) to better understand their effects. Second, the content used for knowledge assessment is relatively basic. Future work will expand both the scope and diversity of the questions. In addition, we did not compare voice-only AI agent with embodied AI agent in unstructured tasks in this study, as participants were already familiar with voice-based interactions from earlier structured tasks. We also did not include a fully human-operated agent as a baseline. A more comprehensive comparative study would help clarify the nuanced differences in unstructured tasks and further explore the gap between the AI agent and human interaction. Finally, as with many Wizard-of-Oz setups, the reliance on manual observation may introduce slight variations in agent behavior, potentially affecting the consistency and accuracy of proactive responses. Making the system fully

automated could make it more reliable and easier to scale in future deployments.

REFERENCES

[1] C. Tikva and E. Tambouris, "Mapping computational thinking through programming in k-12 education: A conceptual model based on a systematic literature review," *Computers & Education*, vol. 162, p. 104083, 2021.

[2] J.-M. Sáez-López, M. Román-González, and E. Vázquez-Cano, "Visual programming languages integrated across the curriculum in elementary school: A two year case study using "scratch" in five schools," *Computers & Education*, vol. 97, pp. 129–141, 2016.

[3] P. J. Guo, "Older adults learning computer programming: Motivations, frustrations, and design opportunities," in *Proceedings of the 2017 chi conference on human factors in computing systems*, 2017, pp. 7070–7083.

[4] B. Du Boulay, "Some difficulties of learning to program," in *Studying the novice programmer*. Psychology Press, 2013, pp. 283–299.

[5] A. Robins, J. Rountree, and N. Rountree, "Learning and teaching programming: A review and discussion," *Computer science education*, vol. 13, no. 2, pp. 137–172, 2003.

[6] L. E. Winslow, "Programming pedagogy—a psychological overview," *ACM Sigcse Bulletin*, vol. 28, no. 3, pp. 17–22, 1996.

[7] R. P. Medeiros, G. L. Ramalho, and T. P. Falcão, "A systematic literature review on teaching and learning introductory programming in higher education," *IEEE Transactions on Education*, vol. 62, no. 2, pp. 77–90, 2018.

[8] GitHub, "Github copilot," 2021, accessed: 2025-01-18. [Online]. Available: https://github.com/features/copilot

[9] Q. Ma, T. Wu, and K. Koedinger, "Is ai the better programming partner? human-human pair programming vs. human-ai pair programming," *arXiv preprint arXiv:2306.05153*, 2023.

[10] M. Kazemitabaar, J. Chow, C. K. T. Ma, B. J. Ericson, D. Weintrop, and T. Grossman, "Studying the effect of ai code generators on supporting novice learners in introductory programming," in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023, pp. 1–23.

[11] J. Chen, X. Lu, Y. Du, M. Rejtig, R. Bagley, M. Horn, and U. Wilensky, "Learning agent-based modeling with llm companions: Experiences of novices and experts using chatgpt & netlogo chat," in *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 2024, pp. 1–18.

[12] A. M. Dakhel, V. Majdinasab, A. Nikanjam, F. Khomh, M. C. Desmarais, and Z. M. J. Jiang, "Github copilot ai pair programmer: Asset or liability?" *Journal of Systems and Software*, vol. 203, p. 111734, 2023.

[13] I. Wachsmuth, M. Lenzen, and G. Knoblich, "Introduction to embodied communication: why communication needs the body," *Embodied communication in humans and machines*, pp. 1–28, 2008.

[14] P. Romero, B. Du Boulay, J. Robertson, J. Good, and K. Howland, "Is embodied interaction beneficial when learning programming?" in *Virtual and Mixed Reality: Third International Conference, VMR 2009, Held as Part of HCI International 2009, San Diego, CA, USA, July 19-24, 2009. Proceedings 3*. Springer, 2009, pp. 97–105.

[15] M. Brereton and B. McGarry, "An observational study of how objects support engineering design thinking and communication: implications for the design of tangible media," in *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, 2000, pp. 217–224.

[16] A. Merkouris, K. Chorianopoulos, and A. Kameas, "Teaching programming in secondary education through embodied computing platforms: Robotics and wearables," *ACM Transactions on Computing Education (TOCE)*, vol. 17, no. 2, pp. 1–22, 2017.

[17] M. C. Johnson-Glenberg, "Immersive vr and education: Embodied design principles that include gesture and hand controls," *Frontiers in Robotics and AI*, vol. 5, p. 375272, 2018.

[18] A. Larsson and K. Stolpe, "Hands on programming: Teachers' use of metaphors in gesture and speech make abstract concepts tangible," *International Journal of Technology and Design Education*, vol. 33, no. 3, pp. 901–919, 2023.

[19] J. Cassell, J. Sullivan, S. Prevost, and E. Churchill, *Embodied Conversational Agents*, 01 2000.

[20] Z. Ruttkay, C. Dormann, and H. Noot, "Embodied conversational agents on a common ground." 2004.

[21] R. Bovo, S. Abreu, K. Ahuja, E. J. Gonzalez, L.-T. Cheng, and M. Gonzalez-Franco, "Embardiment: an embodied ai agent for productivity in xr," in *2025 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*. IEEE, 2025, pp. 708–717.

[22] Z. Zhao, W. Chai, X. Wang, B. Li, S. Hao, S. Cao, T. Ye, and G. Wang, "See and think: Embodied agent in virtual environment," in *European Conference on Computer Vision*. Springer, 2024, pp. 187–204.

[23] L. A. Williams and R. R. Kessler, "All i really need to know about pair programming i learned in kindergarten," *Commun. ACM*, vol. 43, no. 5, p. 108–114, May 2000. [Online]. Available: https://doi.org/10.1145/332833.332848

[24] L. A. Williams, "Pair programming," in *Encyclopedia of Software Engineering*, 2010. [Online]. Available: https://api.semanticscholar.org/CorpusID:11935089

[25] T. Dybå, E. Arisholm, D. I. Sjoberg, J. E. Hannay, and F. Shull, "Are two heads better than one? on the effectiveness of pair programming," *IEEE Software*, vol. 24, no. 6, pp. 12–15, 2007.

[26] A. Begel and N. Nagappan, "Pair programming: what's in it for me?" in *Proceedings of the Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, ser. ESEM '08. New York, NY, USA: Association for Computing Machinery, 2008, p. 120–128. [Online]. Available: https://doi.org/10.1145/1414004.1414026

[27] S. K. Kuttal, B. Ong, K. Kwasny, and P. Robe, "Trade-offs for substituting a human with an agent in a pair programming context: The good, the bad, and the ugly," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, ser. CHI '21. New York, NY, USA: Association for Computing Machinery, 2021. [Online]. Available: https://doi.org/10.1145/3411764.3445659

[28] Q. Ma, T. Wu, and K. Koedinger, "Is ai the better programming partner? human-human pair programming vs. human-ai pair programming," 2023. [Online]. Available: https://arxiv.org/abs/2306.05153

[29] P. Robe and S. K. Kuttal, "Designing pairbuddy—a conversational agent for pair programming," *ACM Trans. Comput.-Hum. Interact.*, vol. 29, no. 4, May 2022. [Online]. Available: https://doi.org/10.1145/3498326

[30] S. K. Kuttal, J. Myers, S. Gurka, D. Magar, D. Piorkowski, and R. Bellamy, "Towards designing conversational agents for pair programming: Accounting for creativity strategies and conversational styles," in *2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 2020, pp. 1–11.

[31] S. D'Angelo and A. Begel, "Improving communication between pair programmers using shared gaze awareness," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, ser. CHI '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 6245–6290. [Online]. Available: https://doi.org/10.1145/3025453.3025573

[32] E. André, "Design and evaluation of embodied conversational agents for educational and advisory software," in *Handbook of Conversation Design for Instructional Applications*. IGI Global, 2008, pp. 343–362.

[33] B. L. Mencía, D. D. Pardo, A. H. Trapote, and L. A. H. Gómez, "Embodied conversational agents in interactive applications for children with special educational needs," in *Technologies for inclusive education: Beyond traditional integration approaches*. IGI Global, 2013, pp. 59–88.

[34] N. C. Krämer, "Psychological research on embodied conversational agents: the case of pedagogical agents," 2010.

[35] J. Cassell, T. Bickmore, M. Billinghurst, L. Campbell, K. Chang, H. Vilhjálmsson, and H. Yan, "Embodiment in conversational interfaces: Rea," in *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, 1999, pp. 520–527.

[36] L. Zhang and S. Oney, "Studying the benefits and challenges of immersive dataflow programming," in *2019 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, 2019, pp. 223–227.

[37] C. Wee, K. M. Yap, and W. N. Lim, "iprogvr: Design of a virtual reality environment to improve introductory programming learning," *IEEE Access*, vol. 10, pp. 100 054–100 078, 2022.

[38] Z. Zhu, Z. Liu, Y. Zhang, L. Zhu, J. Huang, A. M. Villanueva, X. Qian, K. Peppler, and K. Ramani, "Learniotvr: An end-to-end virtual reality environment providing authentic learning experiences for internet of things," in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, ser. CHI '23. New York, NY, USA: Association for Computing Machinery, 2023. [Online]. Available: https://doi.org/10.1145/3544548.3581396

[39] L. Zhang and S. Oney, "Flowmatic: An immersive authoring tool for creating interactive scenes in virtual reality," in *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, 2020, pp. 342–353.

[40] E. Yigitbas, J. Klauke, S. Gottschalk, and G. Engels, "Vreud-an end-user development tool to simplify the creation of interactive vr scenes," in *2021 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, 2021, pp. 1–10.

[41] H. Ye, J. Leng, P. Xu, K. Singh, and H. Fu, "Prointerar: A visual programming platform for creating immersive ar interactions," in *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, 2024, pp. 1–15.

[42] M. Hedlund, A. Jonsson, C. Bogdan, G. Meixner, E. Ekblom Bak, and A. Matviienko, "Blocklyvr: Exploring block-based programming in virtual reality," in *Proceedings of the 22nd International Conference on Mobile and Ubiquitous Multimedia*, ser. MUM '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 257–269. [Online]. Available: https://doi.org/10.1145/3626705.3627779

[43] R. Oberhauser, "Immersive coding: A virtual and mixed reality environment for programmers," 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:201136604

[44] D. Kao, C. Mousas, A. J. Magana, D. F. Harrell, R. A. Ratan, E. F. Melcer, B. Sherrick, P. Parsons, and D. A. Gusev, "Hack.vr: A programming game in virtual reality," *CoRR*, vol. abs/2007.04495, 2020. [Online]. Available: https://arxiv.org/abs/2007.04495

[45] J. Vincur, M. Konopka, J. Tvarozek, M. Hoang, and P. Navrat, "Cubely: virtual reality block-based programming environment," in *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology*, ser. VRST '17. New York, NY, USA: Association for Computing Machinery, 2017. [Online]. Available: https://doi.org/10.1145/3139131.3141785

[46] S. Viswanathan, B. Omidvar-Tehrani, A. Bruyat, F. Roulland, and A. M. Grasso, "Hybrid wizard of oz: Concept testing a recommender system," in *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020, pp. 1–7.

[47] J. H. Maloney, K. Peppler, Y. Kafai, M. Resnick, and N. Rusk, "Programming by choice: urban youth learning programming with scratch," in *Proceedings of the 39th SIGCSE technical symposium on Computer science education*, 2008, pp. 367–371.

[48] M. A. Novack and S. Goldin-Meadow, "Gesture as representational action: A paper about function," *Psychonomic Bulletin & Review*, vol. 24, pp. 652–665, 2017.

[49] S. Goldin-Meadow, *Hearing gesture: How our hands help us think*. Harvard University Press, 2005.

[50] T. Bickmore, L. Pfeifer, and L. Yin, "The role of gesture in document explanation by embodied conversational agents," *International Journal of Semantic Computing*, vol. 2, no. 01, pp. 47–70, 2008.

[51] F. Wilcoxon, S. Katti, R. A. Wilcox *et al.*, "Critical values and probability levels for the wilcoxon rank sum test and the wilcoxon signed rank test," *Selected tables in mathematical statistics*, vol. 1, pp. 171–259, 1970.

[52] M. Muller, "Curiosity, creativity, and surprise as analytic tools: Grounded theory method," in *Ways of Knowing in HCI*. Springer, 2014, pp. 25–48.

[53] B. Glasser, "Basics of grounded theory analysis: Emergence vs. forcing," *Mill Valley*, 1992.

[54] Y. Schul and N. Peri, "Influences of distrust (and trust) on decision making," *Social Cognition*, vol. 33, no. 5, pp. 414–435, 2015.

[55] P. Kulms and S. Kopp, "The effect of embodiment and competence on trust and cooperation in human–agent interaction," in *Intelligent Virtual Agents: 16th International Conference, IVA 2016, Los Angeles, CA, USA, September 20–23, 2016, Proceedings 16*. Springer, 2016, pp. 75–84.

[56] N. Cha, A. Kim, C. Y. Park, S. Kang, M. Park, J.-G. Lee, S. Lee, and U. Lee, "Hello there! is now a good time to talk? opportune moments for proactive interactions with smart speakers," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 3, pp. 1–28, 2020.

[57] X. Tang, S. Wong, K. Pu, X. Chen, Y. Yang, and Y. Chen, "Vizgroup: An ai-assisted event-driven system for real-time collaborative programming learning analytics," *arXiv preprint arXiv:2404.08743*, 2024.

[58] Q. Jin and C. Borchers, "Who to help? a time-slice analysis of k-12 teachers' decisions in classes with ai-supported tutoring," pp. 1–7, 05 2025.

[59] P. Sykownik, S. Karaosmanoglu, K. Emmerich, F. Steinicke, and M. Masuch, "Vr almost there: simulating co-located multiplayer experiences in social virtual reality," in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023, pp. 1–19.

[60] P. Krop, M. J. Koch, A. Carolus, M. E. Latoschik, and C. Wienrich, "The effects of expertise, humanness, and congruence on perceived trust, warmth, competence and intention to use embodied ai," in *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, 2024, pp. 1–9.

[61] K. Hatalis, D. Christou, J. Myers, S. Jones, K. Lambert, A. Amos-Binks, Z. Dannenhauer, and D. Dannenhauer, "Memory matters: The need to improve long-term memory in llm-agents," in *Proceedings of the AAAI Symposium Series*, vol. 2, no. 1, 2023, pp. 277–280.

[62] D. Aneja, R. Hoegen, D. McDuff, and M. Czerwinski, "Understanding conversational and expressive style in a multimodal embodied conversational agent," in *Proceedings of the 2021 CHI conference on human factors in computing systems*, 2021, pp. 1–10.

[63] C. Obermeier, S. D. Kelly, and T. C. Gunter, "A speaker's gesture style can affect language comprehension: Erp evidence from gesture-speech integration," *Social cognitive and affective neuroscience*, vol. 10, no. 9, pp. 1236–1243, 2015.

[64] A. Najamuddin, "The meaning of gesture in social cultural context," *El-Tsaqafah: Jurnal Jurusan PBA*, vol. 18, no. 1, pp. 102–113, 2019.

[65] D. A. Robb, J. Lopes, M. I. Ahmad, P. E. McKenna, X. Liu, K. Lohan, and H. Hastie, "Seeing eye to eye: trustworthy embodiment for task-based conversational agents," *Frontiers in Robotics and AI*, vol. 10, p. 1234767, 2023.

[66] L. Hoffmann and N. C. Krämer, "Investigating the effects of physical and virtual embodiment in task-oriented and conversational contexts," *International Journal of Human-Computer Studies*, vol. 71, no. 7-8, pp. 763–774, 2013.

[67] R. I. Sutton and A. Hargadon, "Brainstorming groups in context: Effectiveness in a product design firm," *Administrative science quarterly*, pp. 685–718, 1996.