Kashihara, A. et al. (Eds.) (2024). Proceedings of the 32nd International Conference on Computers in Education.
Asia-Pacific Society for Computers in Education

# MESHing Minds: Bridging the Gap Between Creativity and IoT Programming Through Collaborative Mixed Reality

**Yusuke SAKABE[a], Emmanuel AYEDOUN[b*] & Masataka TOKUMARU[b]**
[a]*Graduate School of Science and Engineering, Kansai University, Japan*
[b]*Faculty of Engineering Science, Kansai University, Japan*
*emay@kansai-u.ac.jp

**Abstract:** Fostering creativity in programming tasks is a challenging endeavor, especially when working alone. Traditional programming environments often lack support for stimulating creative thinking and idea generation. This paper presents an interactive augmented reality (AR) system that aims to enhance creativity in Internet of Things (IoT) programming tasks. The proposed system leverages Interactive Evolutionary Computation (IEC) and AR technologies to facilitate the collaborative exploration and evolution of IoT device programs (MESH programs). Users can create, evaluate, and iteratively refine MESH programs through an immersive AR interface, while being inspired by the system's suggestions and other users' creations. The system employs a genetic algorithm to evolve MESH programs based on user evaluations, and utilizes natural language processing to generate program descriptions that can trigger new ideas. A user study (n=16) was conducted to evaluate the system's effectiveness in stimulating creativity and promoting collaboration. Quantitative analysis revealed a significant increase in idea generation over time and a greater impact on inspiration for novice programmers. Qualitative findings highlighted the system's ability to foster a creative and collaborative environment. The insights gained from this study inform the design of future tools and experiences that support creative thinking and collaboration in the IoT domain.

**Keywords:** Augmented Reality, MESH Programming, Collaborative Creativity Support, Internet of Things Ideation, Divergent and Convergent Thinking

## 1. Introduction

In recent years, the proliferation of Internet of Things (IoT) devices has opened up new opportunities for creative expression and problem-solving. By combining various sensors, actuators, and programmable components, individuals can design and develop innovative IoT gadgets tailored to their specific needs and interests. However, creating novel IoT gadget ideas often requires a diverse set of skills, including programming, electronics, and creative design thinking, posing challenges for individuals with limited expertise in these areas.

To address this challenge, we propose an interactive augmented reality (AR) system that leverages the power of Interactive Evolutionary Computation (IEC) and natural language generation to support users in exploring and generating creative IoT gadget ideas. IEC is a human-centric optimization technique that involves users in the evolutionary process, allowing them to guide the search towards desirable solutions based on their subjective evaluations. By incorporating IEC into an AR environment, users can intuitively interact with virtual representations of IoT gadget programs, visualize their behavior, and iteratively refine them through interactive evaluation and selection.

The proposed system utilizes MESH, a visual programming environment for IoT devices, to represent IoT gadget programs. Users can collaboratively create, modify, and evaluate MESH programs using an AR interface, where they can manipulate virtual objects corresponding to MESH components and observe their effects in real-time. To further enhance creativity and understanding, the system incorporates a natural language generation model

that produces descriptive explanations of the generated MESH programs, providing users with additional insights and inspiration.

The main contributions of this work are:

1. An interactive AR system that leverages IEC and natural language processing to support collaborative exploration and evolution of IoT device programs.
2. A user study evaluating the system's effectiveness in fostering creativity and its potential to support collaborative programming experiences.
3. Insights and recommendations for designing interactive AR systems for creative programming tasks.

## 2. Related Works

### *2.1 Creative Thinking*

Creative thinking is a cognitive process that involves generating novel solutions to problems without relying solely on prior knowledge or experience (Guilford, 1959). This process consists of two essential components: divergent thinking and convergent thinking. Divergent thinking refers to the generation of diverse ideas without being constrained by preconceived notions, allowing for flexible and open-ended exploration of possibilities. Convergent thinking, on the other hand, involves selecting and synthesizing the most useful ideas from the generated pool to formulate effective solutions to the problem at hand. The interplay between divergent and convergent thinking enables individuals to approach a given issue from multiple perspectives and seek innovative solutions through the generation and refinement of ideas.

The importance of incorporating others' perspectives in creative thinking has been highlighted in numerous studies. In the field of cognitive science, researchers have suggested that projecting the ideas of others with different viewpoints onto oneself can facilitate serendipitous discoveries and lead to creative solutions (Oehlmann, 2003). This notion has been further explored in the context of sketching tasks, where the access to others' viewpoints has been shown to support creative thinking (Miell et al., 2004; Sawyer, 2012). By incorporating the perspectives of others, individuals can uncover novel ideas and insights that they might not have conceived independently, thus expanding their creative potential.

Despite the significance of both divergent and convergent thinking in creative problem-solving, most existing idea support systems tend to focus on only one aspect of the creative thinking process. Few studies have investigated the effective combination of both divergent and convergent thinking in idea support systems. Moreover, while some previous idea support systems have incorporated others' viewpoints, they have not fully exploited the potential of leveraging the broadened perspectives of peers to generate new ideas and support creative thinking.

### *2.2 Technology Enhanced Creativity and Ideation Support*

Creativity support systems have been a focus of research in various domains, aiming to enhance and facilitate creative processes. These systems provide tools, techniques, and environments that stimulate ideation, encourage exploration, and assist users in generating novel solutions. Several studies have investigated the effectiveness of creativity support systems. Shneiderman et al. (2006) proposed a framework for designing creativity support tools, emphasizing the importance of easy exploration, rich history-keeping, and collaboration support. Similarly, Resnick et al. (2005) outlined design principles for tools that support creative thinking, focusing on the need for low thresholds, high ceilings, and wide walls in creative environments. Frich et al. (2019) conducted a survey of creativity support tools and identified common features such as inspiration galleries, sketching interfaces, and generative algorithms.

There have been several approaches to support creative thinking and idea generation. A notable one is the Interactive Evolutionary Computation (IEC) approach, where users interact with a system to evolve solutions through iterative evaluation and selection (Takagi, 2001). IEC has been successfully employed in fields such as graphic design (Secretan et al., 2011), music composition (Jónsson et al., 2015),-user interface design (Quiroz et al., 2007) and 3D modeling (Clune and Lipson, 2011). These applications demonstrate the potential of IEC to generate novel and personalized solutions based on user preferences and feedback. By involving users in the evolutionary process, IEC can capture subjective criteria and adapt to individual creative goals.

Augmented Reality (AR) has also been explored for supporting creativity and idea generation. AR overlays digital information onto the real world, providing immersive and interactive experiences that enhance understanding and exploration. For instance, ElSayed et al. [2016] proposed an AR system to support idea exploration during the early design phase. Cascini et al. (2018) developed an AR-based system for co-creative design, allowing users to manipulate virtual objects and collaborate in real-time. Suzuki et al. (2020) proposed an AR-based collaborative sketching tool that enables multiple users to create and share ideas in a shared virtual space. These examples highlight the potential of AR to facilitate collaboration and creative exploration in various domains through enhanced spatial awareness and in-situ visualization.

### *2.3 Collaborative Programming Environments*

Collaborative programming environments have been developed to support teamwork, knowledge sharing, and collective problem-solving in software development. Existing collaborative programming environments incorporate features such as real-time code editing, version control, and communication tools. For example, Collece (Bravo et al., 2013) is a collaborative coding environment that supports real-time collaboration, code review, and issue tracking. These tools aim to facilitate teamwork and improve the efficiency of collaborative programming tasks. On the other hand, several AR-based programming tools have been developed to assist learners in understanding and visualizing programming concepts. Narman et al. (2020) created an AR-based learning environment for teaching data structures and algorithms, using interactive visualizations to illustrate complex concepts. Sittiyuno and Chaipah (2019) developed ARCode, an AR application aiming to help beginners to understand the workflow of a program, and provide a collaborative learning environment. These tools demonstrate the potential of AR to enhance programming education by providing intuitive and engaging learning experiences.

While existing collaborative programming environments have shown promise in supporting teamwork, there are opportunities to enhance these tools to support collaborative creative thinking by integrating AR and generative AI technologies. AR can provide immersive and interactive visualizations of code structures, fostering a shared understanding among collaborators. On the other hand, generative AI techniques such as natural language processing and computer-driven optimization techniques such as IEC, can assist in generating code descriptions, suggestions, identifying potential bugs, and facilitating communication between team members.

## 3. Proposed Environment

### *3.1 Problem Statement and Novelty of Approach*

Existing idea support systems often focus primarily on either divergent or convergent thinking, failing to effectively integrate both aspects of the creative process. Moreover, while some systems have incorporated others' viewpoints to stimulate creative ideation, they have not fully leveraged the potential of harnessing diverse perspectives to generate novel ideas and support creative problem-solving.

Our proposed AR-based creativity support system introduces a dual-source approach to stimulating creative thinking. By presenting users with both system-generated programs, based on their preferences, and programs created by their peers, the system encourages creative ideation from two distinct sources: the system's algorithmic suggestions and the innovative and relevant ideas of fellow users. This unique integration sets our system apart from existing approaches, as it effectively combines divergent and convergent thinking while harnessing the power of diverse perspectives to support creative problem-solving.

The proposed system's ability to support the creation, visualization, and modification of MESH programs, while providing computer-generated suggestions and descriptions, aligns with the principles of easy exploration, rich history-keeping, and collaboration support. By offering a seamless and intuitive interface for exploring and refining IoT programming ideas, the system lowers the threshold for engagement and encourages creative experimentation.

## *3.2 System Features*

The proposed system, as described above, aims to provide an immersive and interactive environment that stimulates creative thinking and idea generation in IoT programming tasks. The key features of the system are described below.

### *3.2.1 Visual Programming*

MESH is an innovative programming platform developed by Sony that enables users to create interactive projects and explore computational thinking concepts through a series of engaging challenges. The platform consists of various types of physical blocks (Figure 1-Left), such as LED, Button, Motion, Move, Brightness, Temperature & Humidity, and GPIO, which can be programmed using a simple drag-and-drop interface in the MESH app. By connecting these blocks and specifying their behaviors, users can create a wide range of interactive projects, from smart light bulbs to security cameras.

MESH's visual programming approach and modular design make it well-suited for Internet of Things (IoT) programming and creative exploration. The intuitive interface allows users to focus on the high-level logic and functionality of their IoT applications, reducing the cognitive load associated with traditional text-based programming. The pre-defined blocks and simulation capabilities enable rapid prototyping and experimentation, fostering creativity and innovation in IoT solution development. Additionally, MESH's compatibility with a wide range of IoT devices and protocols makes it a versatile platform for creative exploration in various IoT domains.

MESH programming emphasizes key computational thinking principles, including sequencing, decomposition, and understanding the IoT. Through a series of increasingly complex challenges, users learn to break down problems into smaller, more manageable components and create programs that combine simple sequences to achieve desired outcomes. The MESH platform not only provides a hands-on approach to learning programming concepts but also encourages creativity and innovation by allowing users to design and build their own unique projects. Figure 1 (Right) shows a MESH program which


*Figure 1. Left:* MESH blocks and visual programming user interface, *Right:* Example of simple program for MESH-based IoT application for motion-triggered camera and LED activation

demonstrates an Internet of Things application where detecting motion causes a camera to capture an image and an LED to illuminate.

#### *3.2.2 AR Interface*

The AR interface provides an immersive and interactive environment for users to visualize, manipulate, and evaluate the generated IoT application designs. To implement it, we leverage the capabilities of the Microsoft HoloLens 2, a state-of-the-art AR headset. It offers advanced features such as spatial mapping, hand tracking, and gesture recognition, enabling intuitive user interactions with virtual objects. Users interact with the AR interface using natural hand gestures. The generated MESH programs are visualized as 3D holograms, allowing users to examine the structure and connectivity of the IoT application designs. Users can manipulate the holograms, rearrange components, and make modifications to the designs using intuitive gestures such as grabbing, moving, and scaling.

Moreover, the AR interface is seamlessly integrated with the MESH programming platform. In other terms, the changes made to the designs in the AR interface are synchronized with the underlying MESH code, enabling users to experiment with different configurations and observe the effects of their modifications in real-time. This further ensures consistency between the visual representation and the actual program logic. This hands-on approach is expected to help fostering creativity and facilitating a deeper understanding of IoT programming concepts.

#### *3.2.3 Natural Language Generation*

The natural language processing module, powered by GPT-3.5, is incorporated into the proposed system to generate program descriptions and provide explanations of the generated IoT application designs. GPT-3.5 is employed to generate human-readable descriptions of the MESH programs evolved by the IEC algorithm. The generated descriptions aim to provide users with a high-level understanding of the functionality and purpose of each IoT application design. GPT-3.5 takes the MESH code as input and generates coherent and informative descriptions using its language understanding capabilities. These descriptions aim to trigger new ideas and perspectives among users, further stimulating creative thinking. To ensure the quality and relevance of the generated descriptions, prompt engineering techniques are applied. The prompts used to query GPT-3.5 are carefully crafted to guide the model towards generating descriptions that are specific to the IoT domain and the MESH programming language.

### *3.3 System Implementation*

The proposed system's technical architecture, as illustrated in Figure 2, comprises several interconnected components that facilitate collaborative programming and creative exploration using MESH blocks. At the core of the system is a web server hosting Streamlit, a Python framework for developing interactive web applications. Streamlit serves as the user interface for writing descriptions of MESH programs, allowing users to input and share their ideas seamlessly. The web server communicates with a host server via HTTP to transmit user-generated description data. The host server is tasked with critical functions such as FastAPI-based HoloLens synchronization management, users' description management, and HoloLens downtime recovery, ensuring smooth operation and data consistency across the system.

Users interact with the system through their PCs, which display the Streamlit-based user interface for writing descriptions of their MESH programs. The MESH blocks themselves are equipped with sensors that collect data from the surrounding environment. This sensor data is transmitted to Unity, a powerful game development platform that plays a crucial role in the system's IEC process. Unity utilizes the sensor data and user preferences to generate MESH programs that align with users' creative intent. The generated MESH programs are

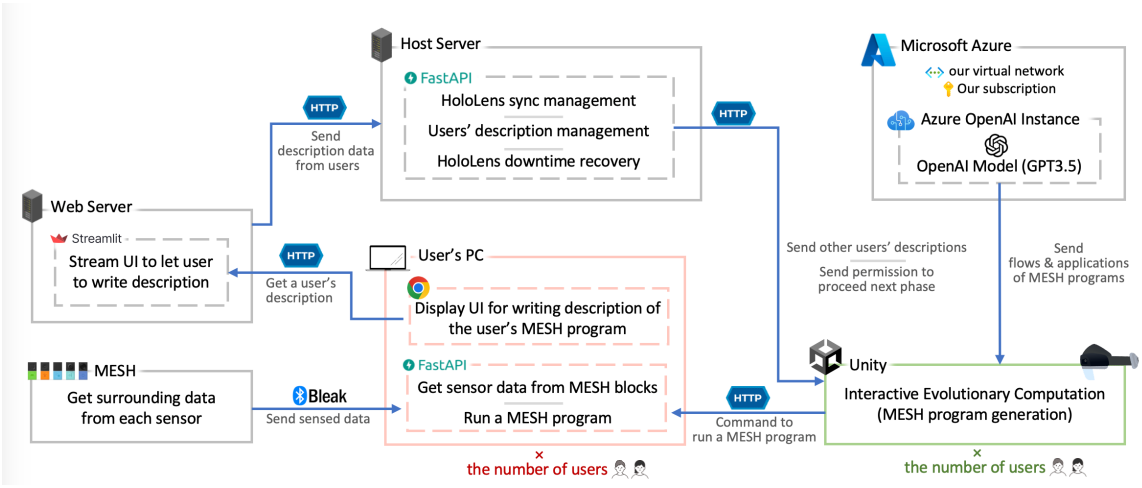


*Figure 2.* The technical architecture of the proposed system, showcasing the interconnected components that enable collaborative programming and creative exploration using MESH blocks.


*Figure 3. Top:* The evaluation phase interface, where users can view and provide feedback on MESH programs created by other users or generated by the system. *Bottom:* The programming phase interface, where users can create MESH programs by connecting virtual MESH blocks in an AR environment.

then managed by the main server, which oversees their flow and applications within the system.

We rely on the robust Microsoft Azure platform, leveraging its Azure OpenAI Instance and the OpenAI GPT-3.5 model for natural language processing tasks. This integration allows

the system to interpret and process user descriptions effectively, enhancing the overall user experience and enabling more accurate generation of MESH programs.

### *3.4 Interaction Flow*

The system supports a collaborative programming experience by allowing multiple users to participate in the MESH program evolution process simultaneously. Users are provided opportunities to view and evaluate each other's MESH programs, fostering indirect idea exchange and collective exploration. Users' interaction within the system consists of two main phases: the evaluation phase, which promotes divergent thinking, and the programming phase, which encourages convergent thinking.

#### *3.4.1 Evaluation Phase*

In the evaluation phase, the system presents MESH programs created by other users or generated by the system itself. Users can access descriptions corresponding to the presented MESH programs, which outline the program flow and provide application examples, as illustrated in Figure 3 (Top). Users can rate or provide feedback on these programs, which is then used by the IEC algorithm to generate new program candidates.

By presenting ideas from the system and other users, the system stimulates the users' divergent thinking and encourages the generation of new ideas.

#### *3.4.2 Programming Phase*

During the programming phase, users create new MESH programs based on the ideas acquired in the evaluation phase. The programming interface provides an AR environment where users can construct MESH programs by connecting virtual MESH blocks. As shown in Figure 3 (Bottom), the interface displays various MESH components, such as sensors and actuators, which can be combined to create IoT applications. Users can select physical objects or devices from their surroundings and associate them with the virtual MESH components, enabling the creation of IoT gadgets that interact with the real world. After a user creates a MESH program, the system presents suggestions of other MESH programs that share common elements with the user's creation, further supporting idea generation. By providing an opportunity to transform ideas into tangible programs, the system promotes convergent thinking among users.

The evaluation and programming phases are repeated for a predetermined number of iterations. It is important to note that the system shares the MESH programs and descriptions created during the programming phase with other users who are simultaneously using the system. The shared MESH programs and descriptions, along with those generated by the system, are then presented to users in the evaluation phase.

This cyclical process of evaluating and programming, combined with the sharing of user-created content is designed to foster a collaborative and creative environment where users can draw inspiration from both the system's suggestions and the ideas of their peers, ultimately enhancing the ideation and development of innovative IoT applications.

## 4. Experimental Evaluation

### *4.1 Study Design and Procedure*

To evaluate the effectiveness of the proposed multi-participant creative support system, a controlled user study was conducted. The study employed a within-subjects design with counterbalancing to minimize order effects. A total of 16 university students (10 males and 6

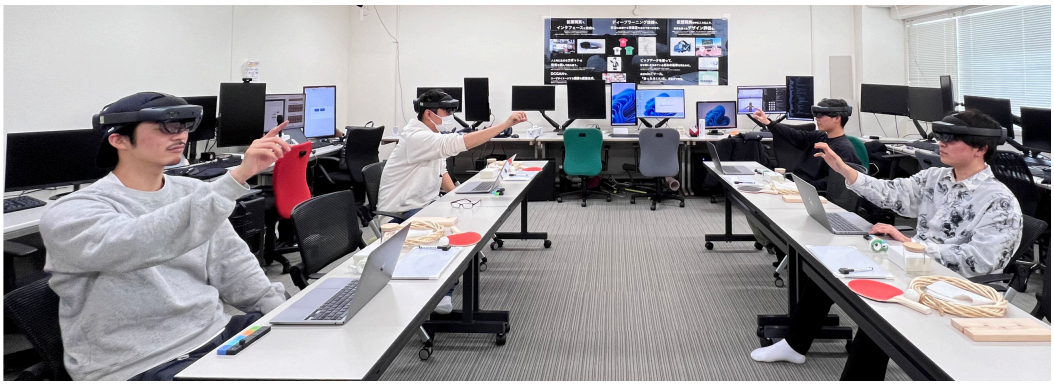

*Figure 4.* Experimental setup for the study showing a group of four participants

females) aged between 18 and 24 years (M = 20.5, SD = 1.8) were recruited from diverse academic backgrounds, including computer science majors ranging from bachelor's to master's levels. Before the study, participants completed a questionnaire to assess their programming experience, based on which they were categorized as novice, intermediate, or expert programmers. These 16 participants were then divided into 4 groups of 4 participants each.

Each group participated in two 60-minute sessions, separated by a 15-minute break. The experiment was designed to consist of multiple iterations of the evaluation and programming phases, with each complete cycle allocated a specific duration. The time allocated for each phase within a cycle was carefully considered to ensure that participants had sufficient opportunities to engage in both divergent and convergent thinking processes. During each session, participants wore HoloLens 2 AR headsets and were asked to complete a series of creative programming tasks using the provided system. The programming tasks involved visual programming by selecting and connecting MESH blocks to create unique IoT gadgets using real-world objects, such as a ping-pong paddle, a ping-pong ball, a whiteboard, a wooden block, a rope, a mirror, an adhesive tape, etc.. Participants were encouraged to think creatively and explore various possibilities to design IoT gadget with these objects using the proposed system. Figure 4 shows the experiment setting featuring a group of four participants.

To assess the system's impact on creative thinking, participants completed questionnaires before and after each session. The questionnaires measured various aspects of creativity, such as idea generation, originality, and inspiration using a 5 point Likert-Scale. Semi-structured interviews were also conducted to gather qualitative insights into participants' experiences and perceptions.

### *4.2 Results*

Quantitative analysis of the questionnaire responses revealed significant effects of the proposed system on participants' creative thinking. A repeated-measures ANOVA showed a significant main effect of time (before vs. after the break) on the number of MESH programs created ($F(1, 15) = 7.2$, $p < 0.05$, $\eta^2 = 0.32$). As illustrated in Figure 5-Left, participants created more programs after the break (M = 8.0, SD = 1.3) compared to before (M = 5.8, SD = 1.1), indicating an increase in idea generation over time.

Additionally, a significant interaction effect was found between programming experience and the system's impact on inspiration ($F(2, 13) = 4.8$, $p < 0.05$, $\eta^2 = 0.42$). Post-hoc tests revealed that novice programmers (M = 4.2, SD = 0.4) experienced a significantly greater increase in inspiration compared to intermediate (M = 3.7, SD = 0.5) and expert programmers (M = 3.2, SD = 0.4), suggesting that the system was particularly beneficial for those with less programming experience (Figure 5-Right).

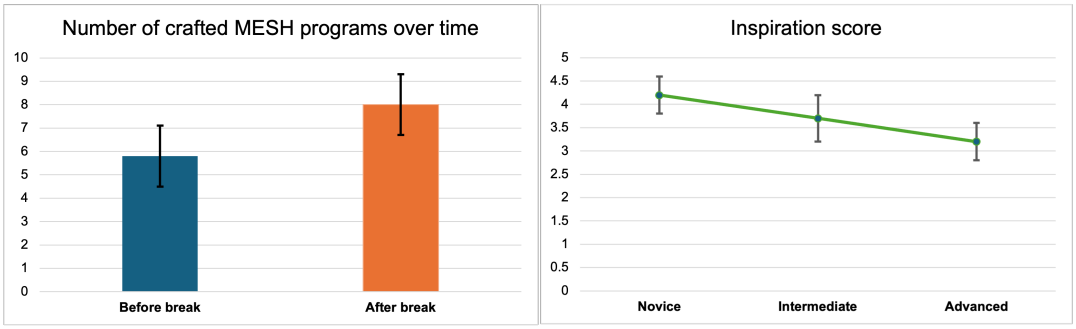


*Figure 5.* Left: Main effect of time on the number of MESH programs created Right: Interaction effect between programming experience and the system's impact on inspiration

Qualitative analysis of the interview data provided further insights into participants' experiences. Many participants praised the system's ability to stimulate creative thinking, with one stating, "*The AR environment and the presented programs really helped me think outside the box and come up with ideas I wouldn't have considered otherwise.*" Others appreciated the collaborative aspect, noting that "*Seeing other people's creations and getting feedback on my own programs was really motivating and inspiring*."

However, some participants also identified areas for improvement. Several novice programmers found the task allocation times to be too short, with one explaining, "*I sometimes felt rushed and couldn't fully explore all the possibilities.*" Expert programmers, on the other hand, suggested increasing the complexity and flexibility of the available MESH blocks to allow for more advanced creations.

### *4.3 Discussion and Conclusion*

The findings of this study provide preliminary evidence for the effectiveness of the proposed multi-participant creative support system in promoting creative thinking in visual programming tasks. The observed increase in idea generation over time suggests that the system's evolutionary algorithm and collaborative features may have contributed to stimulating participants' creativity. Additionally, the finding that novice programmers experienced the greatest increase in inspiration highlights the potential of the system to support and encourage individuals who are new to programming.

The qualitative data gathered from the interviews offer further insights into the system's ability to create a creative and collaborative environment. Participants' positive feedback regarding the AR interface and the exposure to diverse perspectives is consistent with prior research on the benefits of immersive technologies and social creativity support tools. However, it is important to acknowledge that the small sample size of this study limits the generalizability of these findings. Future research with larger and more diverse participant groups is necessary to validate and extend these initial results.

The study also identified several areas for future system development and improvement. The pre-study questionnaire revealed varying needs and preferences among participants with different levels of programming experience, suggesting that incorporating adaptive task allocation and difficulty adjustment could enhance the system's effectiveness and user experience. Moreover, the inclusion of more advanced MESH blocks and greater flexibility in program creation could better cater to the needs of expert programmers while still providing support for novices.

Despite the limitations of the small sample size, the study's findings have potential implications for the design of creative support tools in programming education and practice. The promising results regarding the system's ability to promote creative thinking and collaboration highlight the potential of combining AR and evolutionary computation to foster innovation and problem-solving skills. However, further research is needed to investigate the

long-term effects of such systems on programmers' creative development and to explore their application in real-world contexts, such as educational institutions and software development teams.

In conclusion, while this study provides initial evidence for the effectiveness of the proposed multi-participant creative support system in promoting creative thinking in visual programming tasks, the small sample size limits the generalizability of the findings. Further research with larger and more diverse participant groups is necessary to validate and extend these results. Nonetheless, the insights gained from this study offer valuable direction for future system development and highlight the potential of AR and evolutionary computation in supporting creative programming education and practice.

## References


Bravo, C., Duque, R., & Gallardo, J. (2013). A groupware system to support collaborative programming: Design and experiences. *Journal of Systems and Software*, *86*(7), 1759-1771.

Cascini, G., O'Hare, J., Dekoninck, E., Becattini, N., Boujut, J. F., Guefrache, F. B., ... & Morosi, F. (2020). Exploring the use of AR technology for co-creative product and packaging design. *Computers in Industry, 123*, 103308.

Clune, J., & Lipson, H. (2011). Evolving 3d objects with a generative encoding inspired by developmental biology. *ACM SIGEVOlution, 5(4)*, 2-12.

ElSayed, N. A., Thomas, B. H., Marriott, K., Piantadosi, J., & Smith, R. T. (2016). Situated analytics: Demonstrating immersive analytical tools with augmented reality. *Journal of Visual Languages & Computing*, *36*, 13-23.

Frich, J., Nouwens, M., Halskov, K., & Dalsgaard, P. (2021). How digital tools impact convergent and divergent thinking in design ideation. In *Proceedings of the 2021 CHI conference on human factors in computing systems* (pp. 1-11).

Guilford, J. P. (1959). *Traits of creativity*. *In H.H. Anderson (Ed.), Creativity and its Cultivation*, Harper, 142-161

Jónsson, B. Þ., Hoover, A. K., & Risi, S. (2015). Interactively evolving compositional sound synthesis networks. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation* (pp. 321-328).

Miell, D. (2004). Collaborative creativity: Contemporary perspectives. London: Free Association Books Littleton, K. (Eds.)

Narman, H. S., Berry, C., Canfield, A., Carpenter, L., Giese, J., Loftus, N., & Schrader, I. (2020). Augmented reality for teaching data structures in computer science. In *2020 IEEE Global Humanitarian Technology Conference (GHTC)*, 1-7.

Oehlmann, R. (2003). Metacognitive and Computational Aspects of Chance Discovery, New Generation Computing, 3-12, Springer

Quiroz, J. C., Louis, S. J., & Dascalu, S. M. (2007). Interactive evolution of XUL user interfaces. In Proceedings *of the 9th annual conference on Genetic and evolutionary computation* (pp. 2151-2158).

Resnick, M., Myers, B., Nakakoji, K., Shneiderman, B., Pausch, R., Selker, T., & Eisenberg, M. (2005). Design principles for tools to support creative thinking. *National Science Foundation workshop on Creativity Support Tools*. Washington DC.

Sawyer, R.K. (2012). Explaining creativity: The science of human innovation (2nd ed.). Oxford, UK; New York, NY: Oxford University Press

Secretan, J., Beato, N., D'Ambrosio, D. B., Rodriguez, A., Campbell, A., Folsom-Kovarik, J. T., & Stanley, K. O. (2011). Picbreeder: A case study in collaborative evolutionary exploration of design space. *Evolutionary computation*, *19*(3), 373-403.

Shneiderman, B. (2007). Creativity support tools: accelerating discovery and innovation. *Communications of the ACM*, *50*(12), 20-32.

Sittiyuno, S., & Chaipah, K. (2019). Arcode: Augmented reality application for learning elementary computer programming. In Proceedings of *2019 16th International Joint Conference on Computer Science and Software Engineering (JCSSE)* (pp. 32-37).

Suzuki, R., Kazi, R. H., Wei, L. Y., DiVerdi, S., Li, W., & Leithinger, D. (2020, October). Realitysketch: Embedding responsive graphics and visualizations in AR through dynamic sketching. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (pp. 166-181).