A PLAYFUL APPROACH TO EXTENDED REALITY GAME DEVELOPMENT BY INTEGRATING TANGIBLE AND VIRTUAL INTERFACES

Hengbin Zhang, Xinyi Wu, Xiaoyang Wu and Ruowei Xiao
Southern University of Science and Technology, Shenzhen, Guangdong, China

ABSTRACT

Extended reality game technology transcends the boundaries between reality and virtuality, offering an unprecedented gameplay experience. It not only grants game designers and developers with new creative opportunities, but also challenges for traditional game development paradigm. This exploratory study proposes a playful approach that integrates tangible and virtual user interfaces specifically for XR game development. By utilizing experience prototyping and the Wizard-of-Oz experiment, we were able to simulate the proposed hybrid interfaces and gather preliminary user feedback. We further conducted quantitative and qualitative analyses, as well as adopted a rigorous thematic analysis method. Five design implications with generalizable design knowledge were ultimately synergized, with the aim of guiding our subsequent design iterations and inform future game development research within the domain.

KEYWORDS

Extended reality, mixed reality, game development, tangible programming interface

1. INTRODUCTION

Extended reality (XR), as an umbrella term that encompasses Virtual Reality (VR), Augmented reality (AR) and Mixed Reality (MR), refers to spatial, immersive technologies that enable human-beings to observe and interact in a fully or partially synthetic digital environment (Milgram et al., 1995). Specifically, immersive social XR platforms featuring massive multiplayer online video games and open game worlds, are believed not only the literary origin, but also the technical archetype of the hyped term, "Metaverse" (Mystakidis, 2022). Recently, many IT tycoons like Apple, Meta, Microsoft have heavily invested in this sector.

Despite the great potential of XR games, merging physical reality with digital virtuality brings about new challenges for game designers and developers. One of them is that the conventional development paradigm for video games was established with an intensive, if not exclusive, emphasis on the virtual, synthetic contents. Game creators often find themselves busy dealing with all kinds of integration and interoperability issues among heterogeneous physical devices and components, instead of focusing on the design of game mechanism and playful experience per se. Thus, it entails a unified, consistent mechanism to merge, manage and coordinate between diverse physical entities and virtual assets, so as to reduce the technical barriers and development cost. Moreover, many XR games are staged in the application scenarios that require on-site deployment and playtest, e.g. in classrooms, museums, smart factories etc. It indicates the need also for on-the-fly programming and debugging, also known as the "live coding" (Rein et al., 2018).

In this study, we conducted exploratory research to investigate a playful, hybrid approach that combines both virtual and tangible programming interfaces. To be explicit, we first introduced Finite State Machine (FSM), which was also widely adopted in the early game AIs (Sweetser and Wiles, 2002), as a unified model for heterogeneous in-game virtual assets and physical entities, and constructed a compatible experience prototype (Buchenau and Suri, 2000) accordingly. Second, we carried out a Wizard-of-Oz (WoZ) experiment (Dahlbäck, Jönsson and Ahrenberg, 1993) with 25 participants to elicit their feedback regarding the proposed method. The results were analyzed both quantitatively and qualitatively, and synergized into generalizable design implications for guiding our next iteration and possibly future research in this domain.

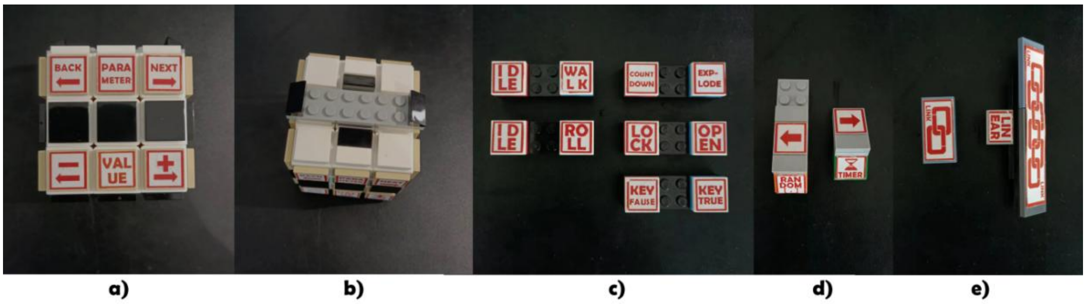

Fig. 1: a) Side view and b) top view of a container. Physical widgets for c) state transfer, d) transfer triggers and e) links.

In the following sections, we will introduce current state-of-the-art research in Section II. Section III outlines the overall research methodology and Section IV presents the processes and outcomes of both quantitative and qualitative data analysis, and V provides a summary of the limitations and contributions.

2. BACKGROUND

The integration of XR technologies with tangible user interface (TUI) has garnered significant interest in the research community in recent years. For example, BlockXR (De Siqueira, Feijóo-García and Stanley, 2021) explores the fusion of physical block programming with projection-based AR and tangible interactions. This integration enables programming actions and feedback to occur seamlessly, thereby minimizing distractions. GeoGate (Ssin et al., 2019) integrates AR with a ring-shaped TUI to facilitate dataset navigation, enabling users to focus on AR content. Similarly, CodeCubes (Cleto et al., 2020) utilizes physical paper cubes as interfaces for manipulating virtual elements, while Draw2Code (Im and Rogers, 2021) and AR-Maze (Jin et al., 2018) combine AR with physical programming blocks, enabling programming in a tangible form and executing generated code within a virtual environment. These approaches provide users with a cohesive view of the operational space, thereby enhancing user understanding and interaction.

Additionally, existing studies have incorporated other technologies to facilitate cross-reality interoperability. For instance, Zanzibar (Villar et al., 2018) introduces a flexible mat interface for tangible user interaction and gaming experiences. It employs NFC and capacitive footprint detection for object tracking, facilitating portability and diverse cross-reality interaction. Similarly, Tlogic (Deng, Wang and Jin, 2019) and Lighters (Wang et al., 2023) utilize image recognition technology to build tangible programming blocks that assist children in problem-solving. Manipulating, placing, and arranging physical objects in space as input allows users to more easily grasp abstract concepts (Li et al., 2022).

The aforementioned studies imply that there is a growing belief among the research community that the hybrid approach can contribute to enhanced novelty, playability and interactivity. Given extant research are mostly education and/or entertainment oriented, there is a rarity of research specifically targeting XR game development tools and the particular challenges that it's confronting. While general-purpose tools like Zanzibar (Villar et al., 2018) emphasize the interactions on a two-dimensional plane, e.g. a desktop, our research distinguishes itself as it explores the design space of three-dimensional spatial interactions and live coding mobility, rendering a layer of playfulness atop the conventional game development approach.

3. METHODOLOGY

We based our study on an existing open framework for Radio Frequency IDentification (RFID) enhanced XR development (Xiao et al., 2022), as our technical foundation. It consists of three major components:

1) A cardboard VR application that runs on smartphones. We opted to utilize the existing Cardboard VR module because it does not rely on any specialized head-mounted display equipment, instead using a smartphone as the display device. This ensures that it can reach a broader user base at a relatively low cost. Meanwhile, we chose to sacrifice some display quality to achieve lighter weight and higher portability, making it suitable for outdoor or other frequently mobile application scenarios.

2) A Real Time Kinematic (RTK) positioning module coupled with Google Map 3D for tracing users' realtime locations and mapping the virtual reality to the actual physical environment.

3) A passive RFID-based interactive module that complements and enhances the restricted interactability of the cardboard VR, which is originally equipped with only one magnetic press button. The RFID module enables a variety of physical-virtual user interactions such as detecting users' body movement, teleporting between different virtual locations, or dynamic loading virtual assets, by simply scanning passive RFID tags.

The above RFID-XR development framework has been proved its technical feasibility and potentials as a novel approach towards cross-reality game genres such as pervasive MR games. However, users currently have to modify a database file to associate RFID tags with their intended operations or effects; for more complicated logic control and functions, user will have to modify or rewrite the source code by themselves. It thus results in insufficient usability and a relatively high technical threshold. To bridge the gap, we hence propose an FSM-based hybrid programming interface that further conceals the underlying technical architecture and facilitates user-defined coordination among both virtual and physical in-game objects.

3.1 Experience Prototype Design

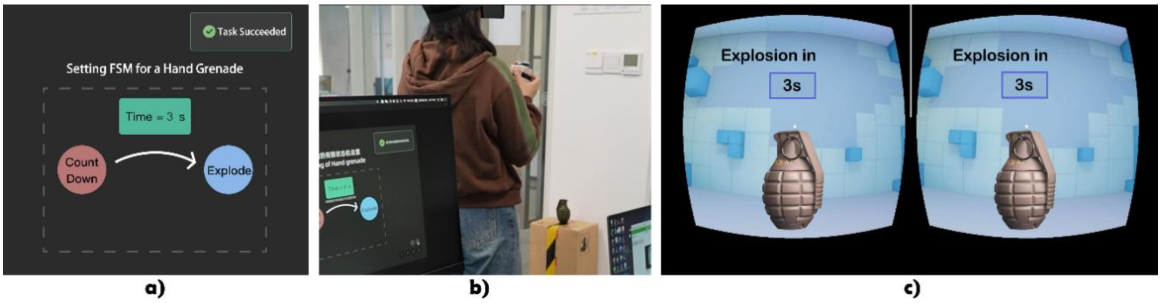

Fig. 2: a) Pseudo-GUI in desktop display. b) A participant used the proposed tool to field debug in front of a physical toy grenade. c) User view in cardboard VR.

We designed our programming interface by referring to the previous research by Xiao et al. (Xiao, Wu and Wang, 2019). The authors devised a domain-specific language for composing IoT devices based on FSM model, from which we adopted some basic notions like the "state transfers" and "state transfer mappings", made them into tangible forms, and extended the targets to both virtual and physical game objects. While programming using a domain specific language, it is highly likely for beginners to create syntax or grammar errors, we applied Token+Constraint (Ullmer, Ishii and Jacob, 2005) method to avoid this kind of obvious bugs, by applying physical constraints like the shape of notch and pluggable widgets in our TUI design.

We adopted an experience prototyping method (Buchenau and Suri, 2000), by utilizing low-tech methods and basic materials to simulate possible interactions and user experience at a relative reasonable cost. Our experience prototype includes both pseudo graphical user interface (GUI) as shown in Fig.2 a), and TUI that consists of several types of tangible components. Together they simulated different use scenarios, including both desktop development as well as field debug/playtest which primarily focuses on fine-tuning of some game parameters and settings. The TUI part was designed based on the concept of "physical analogy of the FSM". Some abstract concepts regarding FSM, e.g. "state transfer", "transfer condition/event" etc., all possess their corresponding physical forms, as shown in Fig.1. Additionally, we also decided to take advantages of fun factors from existing toys like the Rubik's cubes and the LEGO bricks. These playful blocks can not only be easily reused to construct non-functional design prototypes, but also naturally signify the affordances and interactions by drawing upon users' previous experience.

As shown in Fig.1, there are four different types of physical components in our experience prototype:

1) **Containers:** The main body of a container is in the form of a Rubik's cube, which is supposed to accommodate an RFID reader in future stage of functional prototype. A Lego-made notch stays always on its top side, and when a physical widget embedded with an RFID tag is plugged in, a game object will be loaded and its corresponding FSM shown in the GUI. Combined with the feedback from the pseudo-GUI, users can rotate upper row of the cube horizontally for selecting between multiple parameters, while the lower row of the cube for increasing and decreasing the values of the selected parameter. Middle row currently reserved for future use. No vertical rotation is allowed as it affects the position of the top-side notch.

2) **Physical Widgets:** Standard Lego blocks with each representing a pair of state transfer. To specifically note that, the text icons attached to the widgets are only for WoZ experiment purpose. In actual implementation, each physical widget is supposed to be reusable for multiple state transfers.

3) **Transfer Triggers:** A specific type of physical widgets represents the condition or trigger event for a state to transfer to another, e.g. a timer set to three seconds. Also reusable for different conditions and events.

4) **Links:** A specific type of physical widgets for concatenating two or more pairs of state transfer, or two or more FSMs. Additional mapping functions can be attached to a link, e.g. linear, coordinate conversion etc.

3.2 Wizard-of-Oz User Experiment

To gather user feedback about the preliminary design, we conducted a WoZ experiment with 25 participants (16 males, 9 females). In a WoZ experiment, subjects interact with a system that subjects believe to be autonomous, but which is actually being fully or partially operated by a human being (Bella and Hanington, 2012). It was chosen because of our non-functional prototype. Most participants are from fields like design, computer science, and mechanical engineering, ranging from novices to VR/XR experts with over 5 years of experience. The experiment leveraged pseudo-GUIs displayed on both screens and in VR to simulate real-time feedback when the participant performed a task. Each participant was asked to complete three game programming tasks using the proposed prototype within 30 minutes. After completion, the participants filled out a questionnaire and participated in a semi-structured interview. We adhered to the Helsinki Declaration and ethical guidelines from our institution's ethics committee, ensuring informed consent.

During the whole experiment session, the participant was encouraged to apply Think Aloud strategy (Charters, 2003) and required to complete three sequential tasks with a growing complexity. Task I asked the participant to set a grenade from the state "countdown" to the other state "explode", by specifying using a "Timer" type of event trigger. After that, the participant was instructed to don a cardboard VR headset, hold the container, and proceed to a physical location where a toy grenade with embedded RFID tags was placed for field debugging, as shown in Fig.2 b). When participant used the container to scan the RFID tag attached to the grenade, grenade model's appearance, position and parameters will be displayed in cardboard VR view for the participant to further adjust, as shown in Fig.2 c).

Task II asked the participant to switch a door's states between "open" and "locked" using another pair of "true" and "false" states of keys. While the Task III involved the most complicated control logic of establishing a linear mapping between the state transfers of a dice and of a game character, i.e. the state transition of the dice between "idle" and "rolling" will result in randomized dice number, thus triggering the game character's current state to transit from "idle" to "walking" and take the corresponding number of steps.

The participants were asked to respond to the post-experiment questionnaire using a 7-point Likert scale ranging from 1 (strongly disagree) to 7 (strongly agree). We have referred some of our metrics to established assessment indices like Technology Acceptance Model (Davis, 1989) and NASA-TLX (Hart, 2006), while the others were more specific questions related to the proposed hybrid method and FSM model respectively.

4. RESULTS AND ANALYSIS

4.1 Quantitative Analysis Results

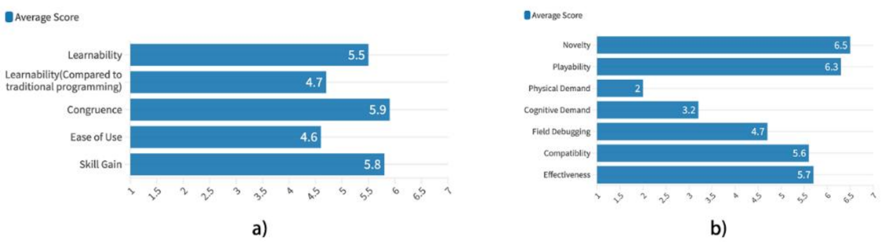

Fig. 3: a) User-perceived overall performance of proposed FSM model. b) User-perceived overall performance of the proposed hybrid user interface.

In this subsection, we present the quantitative analysis results, as shown in Fig.3. To specifically note that, the authors claimed NO statistical significance given the limited sample size. While we acknowledge the discrepancies between the experience prototype used at this stage and the final functional prototype, the primary objective of analyzing quantitative data here, was not to provide rigorous statistical verification but to highlight specific aspects that could guide in-depth follow-up inquiries during subsequent interviews.

According to experimental records, the average completion times for Tasks I, II, and III were 172.44s, 183.88s, and 373.38s, respectively. Observations indicate that participants quickly grasped linking two simple state transfers, resulting in similar times for Tasks I and II. However, Task III required significantly more time due to confusion in mapping state transitions between two different FSMs (dice and game character) using two Rubik's cubes.

Our analysis shows that novices in traditional programming perceived the FSM model more intelligible and learnable. Participants generally found FSM usability as neutral and agreed that FSMs effectively describe behaviors of XR game objects, aiding in understanding these objects and their behaviors in the game.

For the hybrid programming interface, the overall data trend indicates that most participants found it novel and playful. The cognitive and physical burdens during use were minimal, with cognitive load slightly higher than physical load. Most participants affirmed the feasibility of using this tool for field debugging, though a minority were uncertain. Additionally, most agreed that the hybrid interface and its spatial interaction were well aligned with the FSM model.

4.2 Qualitative Analysis Results

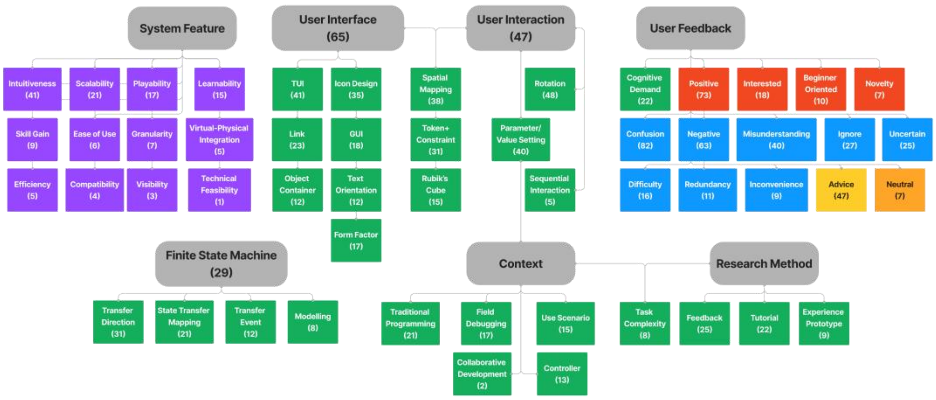

Fig. 4: Affinity diagram showing codes and major themes.

We conducted a thematic analysis based on the qualitative data gathered from our user interview. The interview audio was transcribed into text and proofread by one of the authors manually before distributed to the whole team. All authors then separately coded the first three interviews and established the initial code set through discussions. After the coding scheme was set up, we went through an independent coding process to ensure inter-rater reliability as possible (Braun and Clarke, 2021). All discrepancies emerged from the results of two independent coders were addressed by group discussion to achieve a final agreement over 99%.

The open coding process resulted in a total of 56 codes, and 7 major themes emerged from the codes via affinity diagramming, as shown in Fig.5. Further, by carefully examining the concurrency between codes, we retained and visualized the strongest thematic connections into a chord diagram, as shown in Fig.4. MAXQDA was used as the coding and matrix analysis tool in this process. The authors discussed and highlighted the recurring and unique patterns across different thematic connections, as well as the quotations related to them. Five design implications were extracted and interpreted from the aforementioned thematic analysis results, as elaborated in the following:

1) **Physicalization:** Participants preferred TUI as it enables physicalization (P5, P9, P11, P22, P24) of coding process based on an abstract model, i.e. the FSM, which facilitates embodied interactions and learning, (P4, P20) and intuitive feedback (P4, P19). But some participants also expressed confusion when they realized that a transfer event also is represented by a physical widget. As the participants put it:

P4: "*Compared with just seeing and memorizing it, the tangible interaction might be more effective... It connects the form of LEGO assembly with the underlying programming logic, which I find quite meaningful and intriguing. Traditional LEGO might construct concrete things, but this forms logical connections.*"

P5: "*Because I used to consider the state transfer a very abstract concept, but you have created something very concrete, a physical object corresponding to an abstract concept. I think it is very interesting*."

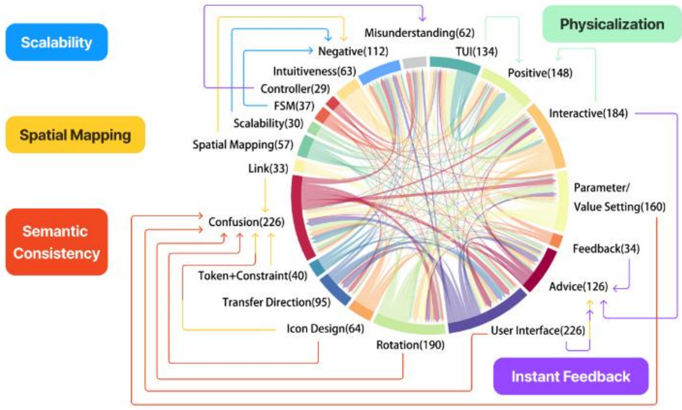

Fig. 5: Chord diagram showing strongest thematic connections.

P11: "*The most interesting part is tangible programming, a process of exploration and discovery using disorganized materials to create novel interactions, feeling like a game with a strong sense of achievement.*"

P10: "*I don't understand what 'the state transition condition named Timer' in the tutorial means; I didn't realize it actually refers to something physical.*"

2) **Semantic Consistency**. Participants expected a consistency between the design of programming interface, the interaction it provides and its associated meaning (P5, P7, P10, P15, P16, P17, P18, P21). Specifically, the visual elements like icons play an important role and offer supplementary instructions to the users, when the token+constraint is not intuitive enough for them to notice (P2, P3, P22, P24). For instance, multiple participants confused the link used in Task II with the one in Task III. Also, some participants expressed the difficulty they encountered when turning the Rubik's cubes, especially when two cubes were connected by a link piece. As the participants put it:

P13: "*Both actions occur on a horizontal plane, and these two actions somehow overlap. but they're quite different—one involves switching parameters, the other adjusting their values. They could be represented using horizontal and vertical axes respectively to distinguish this different.*"

P16: "*When turning the cube, I felt confused and often turned it the wrong way. Despite the stickers, my instinct was that clockwise should increase the value. Also, when trying to turn the top layer, I ended up turning both cubes, likely due to my personal habit of playing with Rubik's cubes*."

P17: "*After using one link piece, I mistakenly thought another link piece was the same, leading to several failed attempts. It was challenging, but that's where I found it valuable and positive*."

3) **Scalability**. As the scale and complexity of the game increase, participants expressed their concerns and uncertainty about both how to utilize FSM for modeling complicated, sequential in-game behaviors (P3, P5, P13), and how users deal with the growing amount of game objects and parameters etc. when scales up (P2, P10, P12, P14). It connects to the user-perceived redundancy, indicating that we should carefully consider the reusability of the physical components when designing and implementing tangible widgets (P19):

P2: "*I'm unsure if the state machine model can handle a vast number of parameters in the future.*"

P3: "*In some XR scenarios, you'll interact with objects whose behaviors can't be easily described by a few parameters, including their reactions to your actions. For example, if I push something, you might install a sensor to measure the force. Then the question becomes how to process and program the related data*."

P19: "*Make it both fundamental and powerful in functionality, using a limited number of basic widgets to achieve various functions than requiring a specific widget for every single task.*"

4) **Spatial Mapping**. A few participants tend to ignore some spatial attributes, e.g. physical slots, form factors and icon design (P7, P8, P19); On the other hand, tangible programming interface in a three dimensional space entails a stricter, more carefully designed token+constraint strategy (P8, P15, P19, P22).

Considering that XR games set in physical venues often require on-site debugging and live coding, the TUI component thus must be portable and mobile, which excludes most desktop or board-based designs. As a three-dimensional object, a Rubik's Cube can fully utilize its form factor as well as spatial motion attributes such as speed, orientation, tilt angle and axis rotation, possibly captured by inertial measurement units (IMU) and other sensors, to provide a sufficiently rich interaction and operation space as a programming tool:

P7: "*First, I believe the exact spatial relationship isn't that important. For physical widgets like Link, I don't mind if the length matches exactly. A spatial mapping doesn't align well with intuition. As long as it's connected, it can be used. This component doesn't seem very intuitive to me.*"

P8: "*Because I haven't seen exactly how the connection structure works, I just thought that as long as they are connected, it would be used. I possibly didn't consider the deeper logic behind it...*"

P13: "*If you're considering using an axis, maybe the icons attached to the Rubik's cube surface aren't necessary. Once you start turning it, it becomes chaotic. Just indicate which side is facing up. Since the center cubelet in each face doesn't change position, you can use them to anchor the directions.*"

5) **Instant Feedback**. Participants usually expected immediate feedback (P1, P5, P9, P12, P17) or traceable history (P24) to facilitate debugging and learning. This kind of expectation or need also to some extent leads to the participants' misunderstanding that the tool acts as a game object controller (P12, P13, P25). Although most participants mentioned about visual feedback, either in conventional GUI or inside VR, the organization and presentation of feedback information should be carefully designed, because elements like texts, icons, pop-ups, menus etc., which traditional 2D GUIs primarily rely on, may cause extra visual fatigue and difficulties in a 3D development environment. Additionally, for hybrid programming approaches, it is still underexploited to further integrate multimodal feedback such as sound and haptics.

P5: "*I wish you could show me the effects of my programming in VR after I set the grenade to explode.*"

P24: "*There could be a step-by-step recording feature that shows how to backtrack. I tried many times just now and can't remember the steps I took. Having a backtrack record would allow for quicker debugging.*"

5. LIMITATION AND CONTRIBUTION

In summary, the first limitation of this study lies in the research method. The authors are fully aware of the inherent constraints of the non-functional prototype and the WoZ user experiment: It is not feasible to offer identical functionality with that of a functional prototype, e.g. manual operations lead to delayed feedback. Also, when compared with other approaches, the FSM model, which we grounded our entire prototype on, has its own pros and cons: while it is a concise and general model that offers cross-reality interoperability, FSM also falls short of scalability, and the programming structure can become overwhelmingly complicated, e.g. the nesting of multiple FSMs, which the TUI design may fail to capture. Particularly in the context of XR games, it may not be suitable for modeling fine-grained and continuous interactions.

As a result, the proposed programming method seems more suitable for targeting game programming education aimed at beginners with no prior experience, or for the composition and playtest with pre-defined FSM templates, such as user-customized XR games for specific application scenarios like museums. More future work can be done with the refinement of the token+constraint design for the tangible interface and related three-dimensional spatial interaction. In our next iteration, the authors plan to implement a fully functional prototype with sensing and actuating modules, e.g. IMU sensors, vibrators etc., with the aim of incorporating richer spatial motion information and multimodal feedback.

As the contribution of this research, we explored a playful, hybrid approach for XR game development, integrating both virtual and tangible user interface. Grounded on an existing open RFID-XR development framework, we further introduced an FSM-based hybrid programming method atop to achieve cross-reality interoperability and embodied, playful interaction, thus to further lower down the technical threshold and development cost for XR game creators, with or without programming skills. By leveraging experience prototyping and WoZ experiment, we were able to collect preliminary user feedback, signaling very positive potentials regarding the novelty and playfulness of the proposed approach. As a result, five design implications were synergized, respectively, physicalization, semantic consistency, scalability, spatial mapping and instant feedback. We expect the overall research outcome to offer novel and informative insights to guide our forthcoming design refinements and future research endeavors in this domain.

REFERENCES

Bella, M. and Hanington, B., 2012. Universal Universal Methods of Design: 100 Ways to Research Complex Problems, Develop Innovative Ideas, and Design Effective Solutions. MA: Rockport Publishers, p.204.

Buchenau, M. and Suri, J.F., 2000, August. Experience prototyping. In: Proceedings of the 3rd conference on Designing interactive systems: processes, practices, methods, and techniques, pp.424-433.

Braun, V. and Clarke, V., 2021. One size fits all? What counts as quality practice in (reflexive) thematic analysis?. Qualitative Research in Psychology, 18(3), pp.328-352.

Charters, E., 2003. The use of think-aloud methods in qualitative research an introduction to think-aloud methods. Brock Education Journal, 12(2).

Cleto, B., Sylla, C., Ferreira, L. and Moura, J.M., 2020. Codecubes: Coding with augmented reality. In: First international computer programming education conference (ICPEC 2020). Schloss-Dagstuhl-Leibniz-Zentrum für Informatik.

Dahlbäck, N., Jönsson, A. and Ahrenberg, L., 1993, February. Wizard of Oz studies: why and how. In: Proceedings of the 1st international conference on Intelligent user interfaces, pp.193-200.

Davis, F.D., 1989. Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS Quarterly, pp.319-340.

Deng, X., Wang, D. and Jin, Q., 2019. TLogic: a tangible programming tool to help children solve problems. In: E-Learning and Games: 12th International Conference, Edutainment 2018, Xi'an, China, June 28–30, 2018, Proceedings 12. Springer International Publishing, pp.255-262.

De Siqueira, A.G., Feijóo-García, P.G. and Stanley, S.P., 2021, October. Blockxr: A novel tangible block-based programming platform. In: 2021 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), pp.1-4. IEEE.

Hart, S.G., 2006, October. NASA-task load index (NASA-TLX); 20 years later. In: Proceedings of the human factors and ergonomics society annual meeting, 50(9), pp.904-908. Sage CA: Los Angeles, CA: Sage publications.

Im, H. and Rogers, C., 2021, June. Draw2code: Low-cost tangible programming for creating ar animations. In: Proceedings of the 20th Annual ACM Interaction Design and Children Conference, pp.427-432.

Jin, Q., Wang, D., Deng, X., Zheng, N. and Chiu, S., 2018, June. AR-Maze: a tangible programming tool for children based on AR technology. In: Proceedings of the 17th ACM Conference on Interaction Design and Children, pp.611-616.

Li, Y., Kothiyal, A., Weber, T., Rossmy, B., Mayer, S. and Hussmann, H., 2022. Designing Tangible as an Orchestration Tool for Collaborative Activities. Multimodal Technologies and Interaction, 6(5), p.30.

Milgram, P., Takemura, H., Utsumi, A. and Kishino, F., 1995, December. Augmented reality: A class of displays on the reality-virtuality continuum. In: Telemanipulator and telepresence technologies, 2351, pp.282-292. Spie.

Mystakidis, S., 2022. Metaverse. Encyclopedia, 2(1), pp.486-497.

Rein, P., Ramson, S., Lincke, J., Hirschfeld, R. and Pape, T., 2018. Exploratory and live, programming and coding: A literature study comparing perspectives on liveness. arXiv preprint arXiv:1807.08578.

Ssin, S.Y., Walsh, J.A., Smith, R.T., Cunningham, A. and Thomas, B.H., 2019, March. Geogate: Correlating geo-temporal datasets using an augmented reality space-time cube and tangible interactions. In: 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), pp.210-219. IEEE.

Sweetser, P. and Wiles, J., 2002. Current AI in games: A review. Australian Journal of Intelligent Information Processing Systems, 8(1), pp.24-42.

Ullmer, B., Ishii, H. and Jacob, R.J., 2005. Token+ constraint systems for tangible interaction with digital information. ACM Transactions on Computer-Human Interaction (TOCHI), 12(1), pp.81-118.

Villar, N., Cletheroe, D., Saul, G., Holz, C., Regan, T., Salandin, O. et al., 2018, April. Project zanzibar: A portable and flexible tangible interaction platform. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, pp.1-13.

Wang, X., Xing, Q., Jin, Q. and Wang, D., 2023. "Be a lighting programmer": Supporting children collaborative learning through tangible programming system. International Journal of Human–Computer Interaction, pp.1-19.

Xiao, R., Vianto, A., Shaikh, A., Hamari, J. and Virkki, J., 2022. Exploring the application of rfid for designing augmented virtual reality experience. IEEE Access, 10, pp.96840-96851.

Xiao, R., Wu, Z. and Wang, D., 2019. A Finite-State-Machine model driven service composition architecture for internet of things rapid prototyping. Future Generation Computer Systems, 99, pp.473-488.