

Situated Game Level Editing in Augmented Reality

Gary Ng¹ Joon Gi Shin¹ Alexander Plopski² Christian Sandor² Daniel Saakes¹
Industrial Design, KAIST¹ Nara Institute of Science and Technology²
Daejeon, Republic of Korea Ikoma, Japan
{garykng,jshin,saakes}@kaist.ac.kr {plopski,sandor}@is.naist.jp

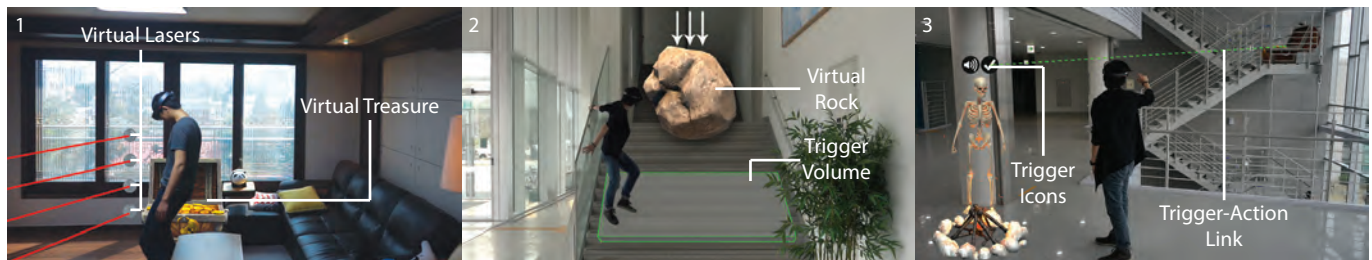


Figure 1. Augmented reality games make use of the unique characteristics of the players' physical space where players must overcome both virtual and physical obstacles (1). Virtual objects, such as a rock, are spatially aware and fall down the stairs when the player enters a trigger volume (2). Our system enables end-users to design personal games with virtual objects and define game logic through spatial trigger-action programming (3).

ABSTRACT

Level editors let end-users create custom levels and content within a given video game. In this paper, we explore the concept and design of Augmented reality game level editors. These new types of editors are not only spatial and embodied, but also situated, as they enable users to tailor games to the unique characteristics and emotional value of their own space.

We present the design and implementation of a prototype level editor that runs on the Microsoft HoloLens. The editor enables users to add virtual content in their homes and add interactions through spatial trigger-action game-logic programming. We had pairs of students create games with the prototype and play each other's games. They reported that games are fun to make, play, and watch others play. Based on the design and evaluation, we propose guidelines for Augmented reality game-authoring tools for end users.

CCS CONCEPTS

• Human-centered computing → user studies; Collaborative Interaction; • Computing methodologies → Mixed augmented reality;

Author Keywords

Augmented Reality; Game Design; Level Editor; Spatial Programming; Trigger-Action Programming

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

TEI '18, March 18–21, 2018, Stockholm, Sweden

© 2018 ACM. ISBN 978-1-4503-5568-1/18/03... 15.00

DOI: <https://doi.org/10.1145/3173225.3173230>

INTRODUCTION

Video game design is a complex task that requires highly technical tools and skills from a variety of fields, including computer science, interaction design, and the arts [20]. Level layout is an important aspect of game design [1, 16] since storytelling and levels guide users through a game. In-game level editors let artists and designers create game content such as layouting game levels by placing objects and scripting interactivity. The key is that these editors do not require the advanced programming skills needed for making the game engine, and so facilitate development. Several games ship such an editor with the game to let players (end users) make their own levels in the game [29]. End-user editors are available for several genres ranging from first-person shooters [53], to strategy games [9], puzzle games [52], and racing games [42]. For other games, such as Super Mario Maker [38] and Minecraft [36], the main activities are content creation and sharing games with other players and in online communities.

In this paper, we look into the design of video game level editors for emerging Augmented Reality (AR) games. Whereas video games currently take place in a virtual environment that is played through a 2D display, games in AR and Virtual Reality (VR) immerse the users in the game and allow for natural and embodied interactions with the virtual content.

Compared to VR, AR games take place in the real world and bring the additional challenge of the game having to adapt to the physical space of the user. For example, a star topology could be applied to a player's living room which turns it into a hub for several challenges in adjacent rooms. A linear story (bus topology) could unfold through the house from the entrance to the stairs and then to the bedroom. In addition, spaces might have emotional significance, such as grandpa's

special chair, and this could be applied to personalize games. That leads to our question: How can we let users personalize or create games that are unique to their living space?

Whereas traditional editors for games support WYSIWYG editing, authoring immersive experiences for VR games challenges the imagination. Therefore, game creation platforms such as Unreal [19] and Unity [50] have started to include level editors that let users design in-situ, in VR. AR game editors bring additional challenges as they are situated in a real space. Therefore, in our design, we address the challenges of 1) in-situ design: The player is in the game when making the game, as opposed to traditional, screen-based level editors that make use of several views such as maps and third and first-person views; and 2) situating and personalizing game content in the user's home. We look specifically at how to create interaction through the use of 3D spatial trigger-action programming (Figure 1).

The contributions of this paper are: 1) the design and implementation of a prototype game level editor for head-worn AR and the considerations that went into it; 2) a formative evaluation with users both playing and designing games; and 3) guidelines for AR game level editors which are useful for game developers. The limitations consists of the small size of the user study that was performed in a laboratory environment instead of the users' homes, the limitations in game content and interaction, and the limitations of using the Microsoft HoloLens.

RELATED WORK

To the best of our knowledge, there is no related work that discusses AR level editors for making interactive games that make use of the user's physical location. The closest is i.Ge [39] in which the virtual game characters interact with ad-hoc manipulation of walls, such as drawing or adding objects.

Content creation in AR differs from other content creation as it adds (or removes) content to the world and there is no need for creating architectural structures like walls or ceilings [48]. Several projects employ Augmented Reality to bring computer games into the real world. Early examples utilized backpack computers and Head-Mounted Displays (HMD) for outdoor games, such as Human Pacman [12] and ARQuake [48], and newer examples employ wearable computers with cameras [27]. Typically, these projects transform gameplay designed for screen and keyboard interaction to spatial user interfaces. Other examples employ Mobile AR [8, 32] on the smartphone, such as Pokemon Go [37].

Although several game genres empower the player with superhuman abilities, there are many design limitations in AR. For example, the game cannot teleport the user back to the start position or require the user to jump off a roof. While changing the user's virtual position in VR could lead to a break in presence, it is not possible in AR. Storytelling techniques relying heavily on controlling the camera, such as cut-scenes, might need new equivalents in AR.

In AR, virtual content can be freely added and moved around, but physical items cannot be rearranged without the aid of human helpers [12] or robotics [41]. Interaction with the physical

environment is also possible through IoT enabled objects, such as opening doors or turning on lights [5]. Everyday objects could function as tangible proxies [23] in the game. In this paper, however, we focus on the layouting and programming of virtual game objects in physical space.

Recent advances in hardware make AR games spatially aware. A few projects [6, 26, 54] employ projector-based systems to situate games in the living room. Fragments [3], a commercial game released for the Microsoft HoloLens, adapts to the user's environment and recognizes objects such as chairs, which allows the virtual characters to sit on them. Situating virtual content to fit to the user's environment is a key challenge of AR gaming that we address in our design. We aim for direct manipulation [47] for object creation and therefore build on work that deals with interior design in AR [13, 30, 31] and 3D content creation [28]. These projects unify action and feedback [17, 49] and employ embodied interaction [18].

Interactivity is important in game design and is typically created in game editors using scripting languages. However, programming in VR [14] or AR [43] suffers from the same problems as programming IoT devices [51] as these interfaces are typically not situated, there is a spatial disconnect between the user and the objects they are working with. To solve this disconnect, some projects [40, 46] and the Reality Editor [24] visualize cause and effect relationships by visualizing virtual connections with directed lines [15]. Another tangible approach to dataflow programming is shown with Spatial Physical Computing [11] in which sensor and actuator modules' relationships are programmed through physical touch. Visual and tangible programming can be applied for beginners or non-technical users to create their own interactive applications [45] and are applied in our design of the AR level editor.

AUGMENTED REALITY LEVEL EDITOR

In this paper, we focus on first-person game experiences [1] in which the player is inside the game world and interacts from the viewpoint of the player's character. This type of game leverages the strengths of the immersive and embodied interactions of head-worn AR (such as the Microsoft HoloLens). Several game genres are available for first-person interactions, such as shooters, construction, and simulations. We selected adventure and puzzle genres to explore the design of an AR editor because they require causal relationship scripting and embed the gameplay in the physical space.

The editor is designed around the concept of visually programming the cause and effect relationships that is common in level editors [7, 29]. For example, as shown in Figure 3, a trigger could be activated when the user enters or exits a space, touches an object, executes a voice command, performs a gesture, or changes their pose. Triggers can cause actions which influence the behavior of non-player characters, start an animation, hide or show objects, or let the user win or lose the game. However, unlike existing end-user programming editors (such as Scratch [44]), we employ situated programming (such as [24, 46]). We define triggers as (collision) volumes in space that are either visible, such as virtual objects, or invisible boxes indicating a space around a real object.

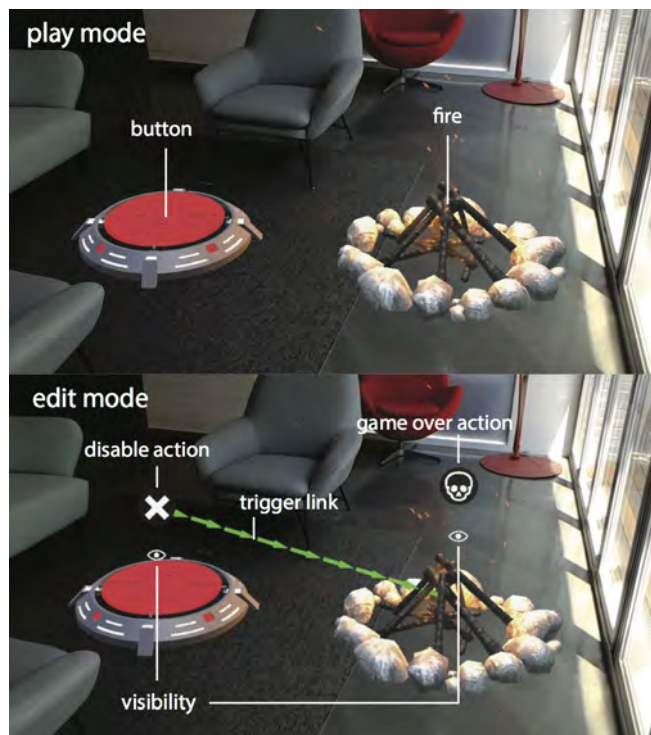


Figure 2. In play mode (top), touching the fire will make the player lose the game, but the fire can be extinguished using the button. In edit mode (bottom), the button has a trigger applied to it which disables the fire. The fire has a game over action applied to it.

AR game editors range from allowing the full authoring of new games to applying templates to an existing space, for instance through the use of wizards. Since, AR games are a new genre, we focus on a basic, but versatile editor, to be employed in a user study. The core functionality of an authoring environment consists of: 1) a library with virtual content and tools to position virtual content in a given location; 2) a scripting environment to program interactivity, non-player character behaviors, and animations; and 3) seamless switching between playing and editing as it allows for explorative design and iteration. In this section we describe the design of a first-person in-situ editor. The editor has an “edit” mode and “play” mode and users switch back and forth with a voice command: “switch mode” to facilitate iterative design with stages of making, testing, and playing. A visual comparison of the two modes is shown in Figure 2. As we focus on the game design process, we base our interaction design decisions on the Apple AR Human Interface Guidelines [2] and the Microsoft mixed reality design guidelines [35] and use a combination of gaze, in-air gestures, and voice commands. Iterative prototyping further guided decisions and helped work around hardware limitations. For instance, because of the narrow Field of View of the HoloLens, we avoided small virtual objects because they are easily missed and reduced the size of the virtual Head Up Display.

Edit Mode

In the edit mode, users design the game by laying out objects and creating relationships between them. Virtual objects can



Figure 3. A treasure chest is set up as a trap. When the player walks over to the treasure chest, it disappears and causes a rock to fall from the sky.

be spawned from a diegetic menu and then selected, moved, rotated, or scaled by performing a selection gesture. Objects that are being moved, are moved relative to the user’s gaze and are aware of the physical environment. Objects slide over the floor and stick on walls (but do not go through walls). Some objects, such as the lasers, automatically adjust orientation to match the normal vector of the surface they are placed on.

We included a library with standard game content. The library consists of three themes (selected for no particular reason): a farm theme with objects such as pigs, haystacks, and a campfire; a pirate theme with skeletons, a treasure chest, and a rock; and a space theme with lasers and spaceships.

Any virtual object can become a trigger (using its bounding box as a volume) and the user can add any number of triggers to the object by staring at it and saying “On Enter” or “On Exit” to create a trigger that responds to whether the user is entering or exiting it. After the trigger type has been assigned, an action must be assigned to it by performing another voice command. We designed four actions, one for winning the game, one for losing the game, one action for showing an object, and one for hiding an object. An example is shown in Figure 2. The button ignites or extinguishes the fire, and touching the fire means “game over”. In edit mode this behavior is scripted with two trigger actions. An **on-enter** trigger is applied to the fire and assigned to a **game over** action. Another **on-enter** trigger is applied to the button with an action to **disable** the fire. A directed line shows the relationship between the button and the fire in edit mode.

Objects can also be set to be disabled when the game starts and need to be activated with an enable trigger in order for them to be shown. Disabled objects do not invoke their triggers, and thus allow for sequential events. Disabling objects is useful for scenarios such as the one shown in Figure 3: a rock becomes visible and falls down when the player reaches the treasure chest.

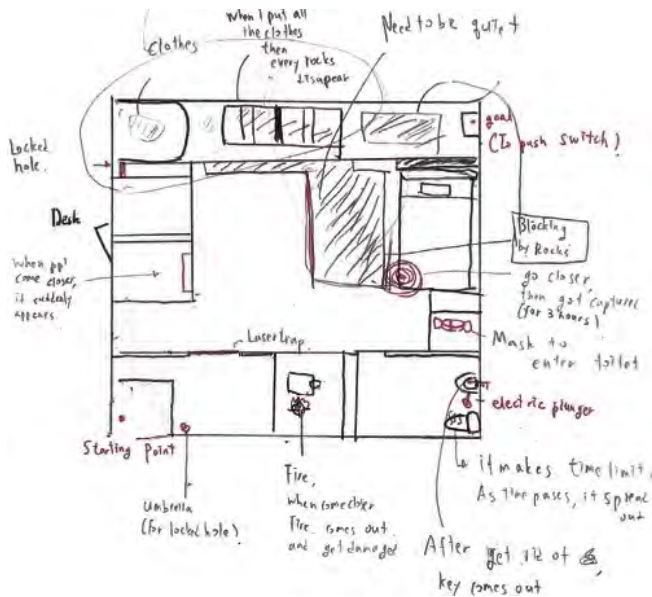


Figure 4. To prepare participants for the user study, we made a sensitizing probe with activities to be completed before the user study. The second-day activity asked to draw a top-down perspective of their space with added game elements. Participant 3A drew an action-adventure game spanning their entire apartment which includes spatial triggers and events.

Play Mode

In play mode, the disabled objects, trigger areas, and UI elements are hidden. The triggers of the enabled objects are also activated, meaning that whenever the user enters or exits an object collider, the linked trigger actions will activate. If the player wants to reset the game, everything will be set back to its initial state when the player says “Reset Game”.

The physical position of the HMD is used to check for collisions with virtual objects in the play mode. When the player enters or exits an object’s collision volume, the triggers are activated, as shown in Figure 3. We defined a $0.5 \times 0.5 \times 2.0\text{m}$ collision volume that extends downwards from the player’s head. This allows for the simulation of the player’s body colliding with virtual objects and also makes it possible for the user to duck below objects. This approximation was necessary because the HoloLens is not capable of tracking the user’s posture and only provides the information about the pose of the user’s head.

EVALUATION

We expect that AR games will be played together within families and with friends. Therefore, in a formative user study we let pairs of participants make games and play each others’ games. From the user study we aimed to discover how participants design games, and how they work with virtual objects in the physical space.

We implemented a prototype as described above. However, during informal tests and a pilot study, we discovered that non-native English speaking participants struggled with issuing voice commands. Since the goal of the study was to explore game play and creation rather than usability, we provided an

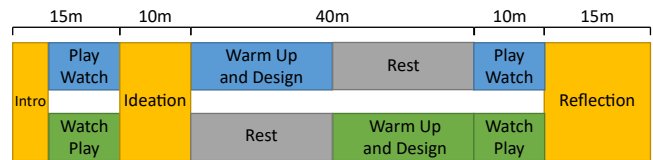


Figure 5. We asked pairs of friends to participate in the user study. After an introduction the pair brainstormed together for ideas. One by one, participants made a game while the other waited in another room. When both designs were completed, participant played each other’s games. The study concluded with a reflection interview.

alternative way of entering voice commands by use of the Wizard of Oz technique [21]. Participants issued voice commands as described earlier, but instead of speech recognition, one of the researchers entered the command on a wireless keyboard. We also noticed that because the voice commands are not self-discoverable, participants needed help during the design session. Therefore, during the study we provided a small card that listed all of the voice commands.

Sensitizing Probe

Because the participants were new to game design, we employed a sensitizing probe [34] to familiarize them with the topic one week prior to the study. The probe consisted of exercises to be executed over two days. For the first-day activity, we tasked the participants to think back to a memorable moment in a recent game they played or a movie they watched and describe how a physical obstacle was overcome and why it was memorable. We designed the second-day activity to get the participants to draw their own space and think about how they would make a game for it. An example of this activity is shown in Figure 4. Participant 3A sketched an adventure game that spanned multiple rooms in his apartment.

All participants completed the probe and they designed a variety of games for their homes. Most games were either of the escape room genre, in which players had to solve a puzzle (e.g. find items) to get out, or action games (e.g. shoot enemies behind a desk, avoiding falling objects). All participants had made the game personal and included elements of their space or physical items they owned. For instance, one participant made a puzzle game in which the player had to find objects in her room that she hid in her desk and wardrobe. The results were not further analyzed but strengthen the idea of personalizing games by situating the game in the user’s space.

Procedure

We recruited four pairs of friends from a local university who did not have game design experience ($n=8$, avg. age=23.25, $SD=1.03$). The study took 90 minutes per pair and we compensated each participant with the equivalent of 9 USD.

The user study took place in a lounge area at a university and attempted to simulate an apartment. The area was L-shaped and contained furniture items that are typically found in living rooms. We set up two video cameras in opposite corners of the room to document the participants’ movements. A laptop captured the first-person video stream from the HoloLens using screen capturing software. We logged users’ actions and movements in Unity (visualized in Figure 7). Adjacent to the

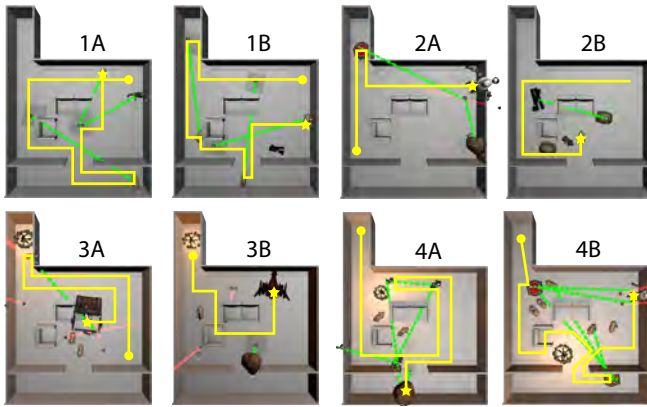


Figure 6. Each of the eight participants made a game during the user study. The green lines are the trigger-action links and the yellow lines represent the path the player was expected to take. The circle indicates the start position and the star marks the goal.

“living room” we created a “design” area with a whiteboard, pens, and paper.

The user study had five phases: introduction, ideation, design, play, and reflection, as shown in Figure 5. The study was designed in a way so that participants could both design and play games. We started by letting the participants play (and watch) an introduction game. During the ideation phase, the participants brainstormed game ideas together without the use of the system.

The design phase was conducted twice. One participant designed while the other rested in a separate room where we provided refreshments. During the design phase, the researcher walked the participant through two warm-up exercises to familiarize them with the interface and functionality. Then, the participants had 20 minutes to design their game.

Once both participants completed their design, they played each other’s games. The designer started by introducing the goal of the game to the player and indicated the starting position of the game. We gave players multiple attempts to beat their partner’s game. We recorded both the player’s and designer’s reactions, as shown in Figure 8.

The final phase of the user study was the reflection phase. We interviewed both participants to discuss their experiences using the system, in both design and play aspects. We based the interview structure on the NASA Task Load Index which was designed to evaluate interface designs and augmented vision [22].

Each session resulted in three video recordings and a user actions and movement log. We synchronized and transcribed the video recordings. We also visualized the design flow of the participants using the log, as shown in Figure 7. Two researchers independently analyzed the probes, transcribed data, videos and metrics and extracted key moments from the study by means of printed screenshots and transcriptions [10]. They identified themes through affinity diagramming [33]. In a second round, the two jointly discussed overarching themes that emerged from the study and are described below.

RESULTS

All of the participants enjoyed designing and playing games and a few participants mentioned that they would like to make games for their family members. Participant 3A said: “I’m going make a game for my parents. For them, our home is more like a sitting place but I want to give it a special meaning for them”. The participants also mentioned how watching someone play their game is also entertaining, especially since they designed the game. Participant 4B said, “When you’re playing this, people can make fun of you and there’s more friendship and interaction... I think they’ll be freaking out but in reality we don’t see anything”. Participant 2B also echoed a similar sentiment, saying, “Since I’m watching her playing the game I wanted to force her to do funny things”.

All participants successfully created games with a victory condition except for participant 1A who forgot to place the victory condition, but notified her partner when she fulfilled it. All games are shown in Figure 6. The eight participants spawned a total of 81 objects (mean = 10.13, SD = 3.09) in total with 59 of them (mean = 7.38, SD = 2.39) present in the final games. Participants mostly used “On Enter” triggers, with 88% (61/69) of all created triggers and 96% (54/56) of all triggers in the final game designs being set to “On Enter”.

None of the participants encountered difficulties playing or making games. Additionally, the NASA TLX and exit interviews did not reveal usability issues. However, the participants frequently mentioned that the in-air gestures in the edit mode were physically demanding and frustrating. Occasionally, when playing the games, touching (or kicking) objects did not trigger events and resulted in a break in presence because objects did not react as expected. This was because the HoloLens only tracks head position and not the user’s limbs. Future HMDs should support full-body tracking to support embodied interaction and natural behavior.

The participants all agreed that the ability to place game objects in a real space was a novel and entertaining task, with participant 4B commenting that the system “was also a bit empowering. I had this kind of fantasy as a kid. Being able to turn your home into a game place, and then, this is exactly what that is”. When participant 1B was playing through the game her partner made for her, she said “it’s actually scary” and was visibly spooked when a skeleton appeared in front of her.

The participants typically made interactions based on the affordance of an object. For example, most of the participants tended to be cautious around items that were perceived to be dangerous, like the lasers. However, since the current prototype allows for the designers to create any trigger on any object, any interaction can be created, even if it is unintuitive. Participant 2A mentioned that she could make an event where “if you touch the pig, a rock falls down. You can make it because the system allows you to do it. But it doesn’t really make sense”. Half of the participants (2A, 2B, 4A, 4B) said that it was difficult to determine the object functionality when playing a game. For example, some games used pigs as a victory condition whereas others had them as dangerous objects.

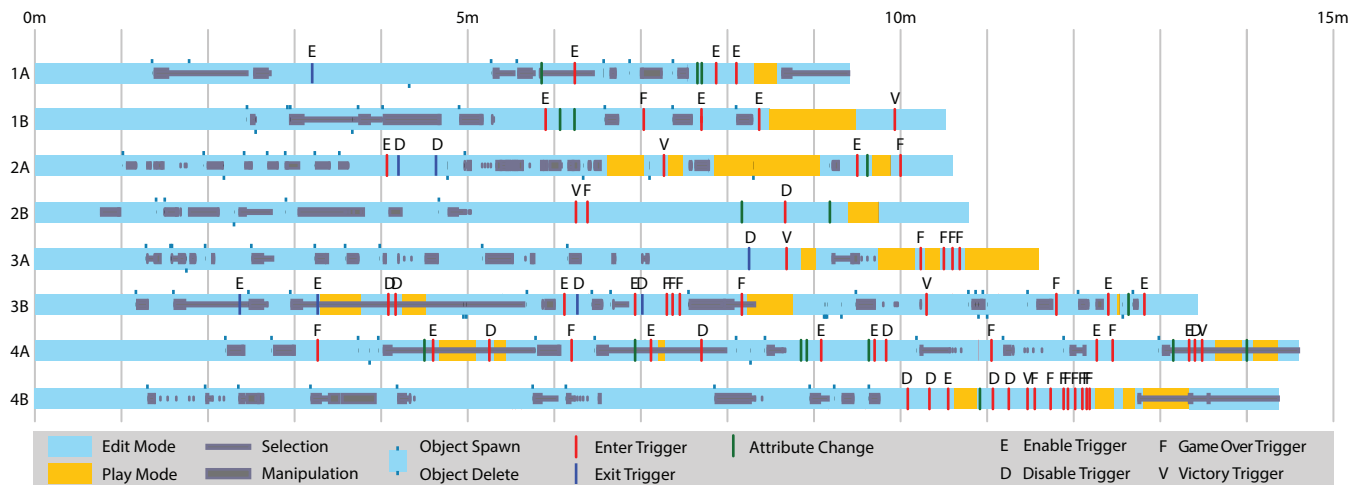


Figure 7. We captured users’ actions during their individual design session to understand their design process. Participants 2A, 3A, 3B, 4A, and 4B utilized an iterative design process by switching between the editor and play mode to test and refine their games, whereas the others first made their entire game before testing it.

Making Use of the Real World

When designing AR games, the participants built their game to fit the physical space. During the ideation phase, two pairs (1 and 4) walked around the physical space to think of game ideas and they commonly used spatial gestures and play acting to express and explain their ideas to their partner. For example, participant 1B walked around the room and pointed out potential walking paths that the player might take and then her partner proposed ideas for where objects should be placed.

The fourth pair also manipulated different physical elements within the space when thinking of ideas. They opened and closed the patio door to explain a trap they could make and even moved some chairs around to define gameplay boundaries and physical obstacles.

The participants expected and enjoyed movement in AR games and so they tried to design games that would include as much physical movement as possible. Participant 3B specifically said that he “*wanted to make it physically challenging so [his partner] had to jump*” so he placed a dangerous object in the corridor between a wall and an armchair which blocked off the path and prompted his partner to try and jump over the object. Participant 1A also mentioned that “*you’re supposed to move. If there was more movement it would be good*”.

Additionally, the participants typically focused on the virtual elements more than the physical environment. Half of the participants (1A, 1B, 3A, 3B) accidentally bumped into something, like the camera or sofa, because they were not paying attention.

Game Design Process

The metrics data, shown in Figure 7, revealed that participants generally take two approaches when making games, an iterative approach alternating editing and playing, and a big-bang approach, designing the entire game followed by a short testing session. Five of the participants (1A, 1B, 2A, 3B, 4A) created the triggers for the objects as they spawned them while the others (2B, 3A, 4B) first layouted all of the objects in the



Figure 8. Several participants mentioned how it was entertaining to not only design and play games, but also to watch their partner play the game they designed.

scene before creating all of the triggers at once. Three of the participants (1A, 1B, 2B) were focused on the act of placing objects and creating triggers and so only tested their games once and did not make many changes based on their testing. However, the other participants (2A, 3A, 3B, 4A, 4B) used an iterative process with the play mode to test their designs and discover logic errors or mistakes, which resulted in relatively more complex and complete games.

All of the participants said that it was not too difficult to create a game, with participant 2A saying that “*you can’t really program other games by yourself*”, referring to traditional game creation methods. However, some participants mentioned that they had difficulty remembering everything they had done or forgot what they needed to do. Participant 1A notably forgot to disable a trap and later mentioned that “*at the end there were so many objects and things that I didn’t think about until [my partner] started playing*” and participant 4A also said something similar, commenting “*sometimes it just tricked my mind because I can’t really remember what I did before so I had to pause and think for a while*”.



Figure 9. AR gaming blends the virtual with the physical. Virtual objects (the laser and pig) were combined with the physical armchair to block the way of the participant so he decided to climb over the armchair to proceed through the game.

DISCUSSION

Based on the design, implementation and evaluation with users, we propose guidelines for the design of future Augmented reality game level editors. We hope that our findings are directly applicable for first-person games and editors and are useful for developers. Usability issues we attributed to the HoloLens (such as speech and gesture recognition, limited Field of View, and body tracking) are excluded from this discussion.

During the user study, we saw participants duck under lasers and jump over both virtual and physical objects, as shown in Figure 9. The participants expected and enjoyed physical activity when playing AR games. Therefore, we propose that future editors and games should **encourage different types of locomotion and actions**, such as jumping, running, and climbing as it would create a more unique and immersive experience that is suitable for AR. However, care needs to be taken to remove unnecessary movement from the game and to highlight areas that could lead to injuries in both modes. In traditional games, characters can be teleported or moved where they need to be, but it is not possible in AR due to the situated context. This means that the understanding of things like the start and end positions in games must be rethought in order to minimize tedious movements.

AR games are played in the real world which means that game content has to “fit” the physical space. Although it is possible to use objects as tangible proxies [23], it is not possible to turn a living room into an expansive desert nor a house into a rabbit hole. Deeper integration is needed with the room’s unique characteristics, perhaps with IoT sensors, so that triggers can be mapped to environmental events. For example, opening a door or moving furniture could serve as a trigger for an in-game event, similar to the connections between virtual objects in the editor.

Some participants found it difficult to start designing since they had no idea what could be done. Game templates that act as tutorials can teach them how to design and assist the users in getting started. Some systems, like Scratch [44], encourage remixing existing programs in order to explore

new ideas [25] and a similar system can be implemented for AR game editors. **Templating functions** could speed up the design process and help new users. Templates could consist of pre-configured modules containing networks of objects and logic can be applied to a physical space and could range from simple interactions, such as buttons that activate doors, to entire games.

Tools and smart guides should be implemented to reduce or eliminate common errors and give users a quick way to holistically view and assess their designs. Game design is a complex task that typically requires several iterations to expose logic errors. In the study we identified a few approaches that users take which indicates that supporting an iterative design process could be beneficial. Additional view modes, such as an overview mode that shows the entire design from a top-down perspective, could also be beneficial to remind users what they have done so far [4].

Augmented reality games should also **facilitate social interaction**. Since AR games are played in context, collocated game modes, such as co-op or competitive modes, are suitable. A “dungeon master” mode, similar to the role of game masters in tabletop role-playing games, can also be implemented where a game master facilitates the game flow using authoring tools while other players play the game. Another avenue would be to let users make games together or in teams.

CONCLUSIONS AND FUTURE WORK

In this paper, we made a first step towards the design of situated level editors in AR that aim to let users design games in their homes. Based on our experience, we provided guidelines for situating the game in the user’s environment and how to use the first-person perspective to program basic interactive content. Our user study indicated that both creation and play can be part of the game experience and suggest multi-user games that make use of both aspects as a next iteration.

The formative user study was performed with only a few users in a laboratory that resembled a living room. Hence, we could not study the intimacy and emotional values of designing in a personal space that are to be expected based on the probe study. Therefore, in the future we intend to perform long-term evaluations “in the wild” in people’s homes where we can investigate how the editor performs on a larger scale with multiple rooms and longer scenarios. Additionally, as the first prototype of its kind, we implemented a subset of the functionality that is required for full-fledged games. Future iterations will include functions such as object and logic grouping and templating. We will also extend the trigger system to include temporal and physiological triggers to enable users to design more complex games. Another avenue that we would like to explore is competitive and collaborative design and play.

ACKNOWLEDGEMENTS

This project was supported by JSPS and NRF under the Japan-Korea Basic Scientific Cooperation Program. This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (NRF-2017K2A9A2A08000200)

REFERENCES

1. Ernest Adams. 2014. *Fundamentals of Game Design, 3rd Edition*. New Riders.
2. Apple. 2017. Human Interface Guidelines: Augmented Reality. (2017). <https://developer.apple.com/ios/human-interface-guidelines/technologies/augmented-reality>
3. Asobo Studio. 2016. Fragments. (2016). <http://www.asobostudio.com/games/fragments>
4. Blaine Bell, Tobias Höllerer, and Steven Feiner. 2002. An Annotated Situation-awareness Aid for Augmented Reality. In *Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology (UIST '02)*. ACM, New York, NY, USA, 213–216. DOI: <http://dx.doi.org/10.1145/571985.572017>
5. Andrea Bellucci, Telmo Zarraonandia, Paloma Díaz, and Ignacio Aedo. 2017. End-User Prototyping of Cross-Reality Environments. In *Proceedings of the Tenth International Conference on Tangible, Embedded, and Embodied Interaction - TEI '17*. ACM Press, New York, New York, USA, 173–182. DOI: <http://dx.doi.org/10.1145/3024969.3024975>
6. Hrvoje Benko, Andrew D. Wilson, and Federico Zannier. 2014. Dyadic Projected Spatial Augmented Reality. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 645–655. DOI: <http://dx.doi.org/10.1145/2642918.2647402>
7. Blender Foundation. 2017. Blender Game Engine: Game Logic. (2017). https://docs.blender.org/manual/en/dev/game_engine/logic/introduction.html
8. Blippar Group. 2009. Layar. (2009). <https://www.layar.com/>
9. Blizzard Entertainment. 2002. Warcraft III. (2002). <http://us.blizzard.com/en-us/games/war3/>
10. Jacob Buur and Astrid Soendergaard. 2000. Video Card Game: An Augmented Environment for User Centred Design Discussions. In *Proceedings of DARE 2000 on Designing Augmented Reality Environments (DARE '00)*. ACM, New York, NY, USA, 63–69. DOI: <http://dx.doi.org/10.1145/354666.354673>
11. Alvaro Cassinelli and Daniel Saakes. 2017. Data Flow, Spatial Physical Computing. In *Proceedings of the Tenth International Conference on Tangible, Embedded, and Embodied Interaction - TEI '17*. ACM Press, New York, New York, USA, 253–259. DOI: <http://dx.doi.org/10.1145/3024969.3024978>
12. Adrian David Cheok, Xubo Yang, Kok Hwee Goh, Wei Liu, Farzam Farbiz, Sze Lee Teo, Hui Siang Teo, Shang Ping Lee, Yu Li, and Siew Wan Fong. 2004. Human Pacman. In *Proc. ACE '04*. ACM Press, New York, New York, USA, 360–361. DOI: <http://dx.doi.org/10.1145/1067343.1067402>
13. Foong-Yi Chia, Bokyung Lee, and Daniel Saakes. 2016. Collaboration in the Living Room or How Couples Design Together. In *Proceedings of the 9th Nordic Conference on Human-Computer Interaction - NordiCHI '16*. 1–6. DOI: <http://dx.doi.org/10.1145/2971485.2996734>
14. Matthew Conway, Steve Audia, Tommy Burnette, Dennis Cosgrove, and Kevin Christiansen. 2000. Alice: Lessons Learned from Building a 3D System for Novices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '00)*. ACM, New York, NY, USA, 486–493. DOI: <http://dx.doi.org/10.1145/332040.332481>
15. R. Dauenhauer and T. Mißjüller. 2016. An Evaluation of Information Connection in Augmented Reality for 3D Scenes with Occlusion. In *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*. 235–237. DOI: <http://dx.doi.org/10.1109/ISMAR-Adjunct.2016.0083>
16. Sjoerd "Hourences" de Jong. 2008. *The Hows and Whys of Level Design - Second Edition*. Self Published.
17. Tom Djajadiningrat, Kees Overbeeke, and Stephan Wensveen. 2002. But How, Donald, Tell Us How?: On the Creation of Meaning in Interaction Design Through Feedforward and Inherent Feedback. In *Proceedings of the 4th Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques (DIS '02)*. ACM, New York, NY, USA, 285–291. DOI: <http://dx.doi.org/10.1145/778712.778752>
18. Paul Dourish. 2004. *Where the action is: The Foundations of Embodied Interaction*. MIT Press.
19. EPIC Games. 2012. Unreal Engine 4. (2012). <https://www.unrealengine.com/>
20. James Paul Gee, Elizabeth Hayes, Robert J. Torres, Alex Games, Kurt R. Squire, and Katie Salen. 2008. Playing to learn game design skills in a game context. *ICLS'08: Proceedings of the 8th international conference on International conference for the learning sciences - Volume 3 3* (2008), 368–374. <http://dl.acm.org/citation.cfm?id=1600040>
21. Paul Green and Lisa Wei-Haas. 1985. *The Wizard of Oz: A Tool for Rapid Development of User Interfaces*. Technical Report. 21 pages. <https://deepblue.lib.umich.edu/bitstream/handle/2027.42/174/71952.0001.001.pdf?sequence=2>
22. Sandra G. Hart. 2006. Nasa-Task Load Index (NASA-TLX); 20 Years Later. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 50, 9 (oct 2006), 904–908. DOI: <http://dx.doi.org/10.1177/154193120605000909>
23. Anuruddha Hettiarachchi and Daniel Wigdor. 2016. Annexing Reality: Enabling Opportunistic Use of Everyday Objects As Tangible Proxies in Augmented Reality. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 1957–1967. DOI: <http://dx.doi.org/10.1145/2858036.2858134>

24. Valentin Heun, James Hobin, and Pattie Maes. 2013. Reality editor. *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication - UbiComp '13 Adjunct* (2013), 307–310. DOI: <http://dx.doi.org/10.1145/2494091.2494185>
25. Benjamin Hill and Andres Monroy-Hernández. 2013. The Cost of Collaboration for Code and Art: Evidence from a Remixing Community. *Collaboration in Creative Communities* (2013), 1035–1045. DOI: <http://dx.doi.org/10.1145/2441776.2441893>
26. Brett Jones, Rajinder Sodhi, Michael Murdock, Ravish Mehra, Hrvoje Benko, Andrew Wilson, Eyal Ofek, Blair MacIntyre, Nikunj Raghuvanshi, and Lior Shapira. 2014. RoomAlive: Magical Experiences Enabled by Scalable, Adaptive Projector-camera Units. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 637–644. DOI: <http://dx.doi.org/10.1145/2642918.2647383>
27. Georg Klein and David Murray. 2007. Parallel Tracking and Mapping for Small AR Workspaces. In *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*. 225–234. DOI: <http://dx.doi.org/10.1109/ISMAR.2007.4538852>
28. Max Krichenbauer, Goshiro Yamamoto, Takafumi Taketomi, Christian Sandor, and Hirokazu Kato. 2014. Towards Augmented Reality user interfaces in 3D media production. In *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 23–28. DOI: <http://dx.doi.org/10.1109/ISMAR.2014.6948405>
29. David Kushner. 2003. *Masters of Doom: How Two Guys Created an Empire and Transformed Pop Culture*. Random House.
30. Manfred Lau, Masaki Hirose, Akira Ohgawara, Jun Mitani, and Takeo Igarashi. 2012. Situated modeling. *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction - TEI '12* (2012), 275. DOI: <http://dx.doi.org/10.1145/2148131.2148190>
31. Bokyoung Lee, Minjoo Cho, Joonhee Min, and Daniel Saakes. 2016. Posing and Acting As Input for Personalizing Furniture. In *Proceedings of the 9th Nordic Conference on Human-Computer Interaction (NordiCHI '16)*. ACM, New York, NY, USA, Article 44, 10 pages. DOI: <http://dx.doi.org/10.1145/2971485.2971487>
32. Blair MacIntyre, Alex Hill, Hafez Rouzati, Maribeth Gandy, and Brian Davidson. 2011. The Argon AR Web Browser and standards-based AR application environment. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*. 65–74. DOI: <http://dx.doi.org/10.1109/ISMAR.2011.6092371>
33. Bella. Martin and Bruce M. Hanington. 2012. *Universal methods of design : 100 ways to research complex problems, develop innovative ideas, and design effective solutions*. Rockport Publishers. 207 pages. <https://books.google.co.kr/books?hl=en>
34. Tuuli Mattelmäki. 2005. Applying probes – from inspirational notes to collaborative insights. *CoDesign* 1, 2 (2005), 83–102. DOI: <http://dx.doi.org/10.1080/15719880500135821>
35. Microsoft. 2017. Design: Learn the Fundamentals of designing mixed reality experiences. (2017). <https://developer.microsoft.com/en-us/windows/mixed-reality/design>
36. Mojang. 2009. Minecraft. (2009). <https://minecraft.net/en-us/>
37. Niantic. 2016. Pokémon Go. (2016). <http://www.pokemongo.com/>
38. Nintendo. 2015. Super Mario Maker. (2015). <http://supermariomaker.nintendo.com/>
39. Patrick Oswald, Jordi Tost, and Reto Wettach. 2014. The real augmented reality: real-time game editor in a spatial augmented environment. *Proceedings of the 11th Conference on Advances in Computer Entertainment Technology - ACE '14* (2014), 1–4. DOI: <http://dx.doi.org/10.1145/2663806.2663853>
40. Heonjin Park, Kyle Koh, Yuri Choi, Han Joo Chae, Jeongin Hwang, and Jinwook Seo. 2016. Defining Rules Among Devices in Smart Environment Using an Augmented Reality Headset. *Proceedings of the Second International Conference on IoT in Urban Space* (2016). DOI: <http://dx.doi.org/10.1145/2962735.2962746>
41. Tetiana Parshakova, Minjoo Cho, Alvaro Casinelli, and Daniel Saakes. 2016. Ratchair. In *ACM SIGGRAPH 2016 Emerging Technologies on - SIGGRAPH '16*. ACM Press, New York, New York, USA, 1–2. DOI: <http://dx.doi.org/10.1145/2929464.2929473>
42. PlayStation Mobile. 2015. GT6 Track Path Editor. (2015). <https://play.google.com/store/apps/details?id=jp.co.polyphony.GTCourseMaker>
43. Iulian Radu and Blair MacIntyre. 2009. Augmented-reality Scratch: A Children's Authoring Environment for Augmented-reality Experiences. In *Proceedings of the 8th International Conference on Interaction Design and Children (IDC '09)*. ACM, New York, NY, USA, 210–213. DOI: <http://dx.doi.org/10.1145/1551788.1551831>
44. Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, and Yasmin Kafai. 2009a. Scratch: Programming for All. *Commun. ACM* 52, 11 (Nov. 2009), 60–67. DOI: <http://dx.doi.org/10.1145/1592761.1592779>
45. Mitchel Resnick, Brian Silverman, Yasmin Kafai, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, and Jay Silver. 2009b. Scratch. *Commun. ACM* 52, 11 (nov 2009), 60. DOI: <http://dx.doi.org/10.1145/1592761.1592779>

46. Christian Sandor, Alex Olwal, Blaine Bell, and Steven Feiner. 2005. Immersive Mixed-Reality Configuration of Hybrid User Interfaces. In *Proceedings of the 4th IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR '05)*. IEEE Computer Society, Washington, DC, USA, 110–113. DOI: <http://dx.doi.org/10.1109/ISMAR.2005.37>
47. B. Shneiderman. 1983. Direct Manipulation: A Step Beyond Programming Languages. *Computer* 16, 8 (Aug 1983), 57–69. DOI: <http://dx.doi.org/10.1109/MC.1983.1654471>
48. Bruce Thomas, Ben Close, John Donoghue, John Squires, Phillip De Bondi, Wayne Piekarski, and Michael Morris. 2000. ARQuake: An Outdoor/Indoor Augmented Reality First Person Application. *Personal and Ubiquitous Computing* 6, 1 (2000), 75–86. DOI: <http://dx.doi.org/10.1007/s007790200007>
49. John Underkoffler, Brygg Ullmer, and Hiroshi Ishii. 1999. Emancipated Pixels: Real-world Graphics in the Luminous Room. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '99)*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 385–392. DOI: <http://dx.doi.org/10.1145/311535.311593>
50. Unity. 2005. Unity. (2005). <https://unity3d.com/>
51. Blase Ur, Elyse McManus, Melwyn Pak Yong Ho, and Michael L Littman. 2014. Practical trigger-action programming in the smart home. *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14* (2014), 803–812. DOI: <http://dx.doi.org/10.1145/2556288.2557420>
52. Valve. 2011. Portal 2. (2011). <http://www.thinkwithportals.com/>
53. Valve. 2012. Counter-Strike: Global Offensive. (2012). <http://blog.counter-strike.net/>
54. Andrew Wilson, Hrvoje Benko, Shahram Izadi, and Otmar Hilliges. 2012. Steerable Augmented Reality with the Beamatron. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*. ACM, New York, NY, USA, 413–422. DOI: <http://dx.doi.org/10.1145/2380116.2380169>