Research Collection School Of Computing and Information Systems

School of Computing and Information Systems

4-2023

Bubbleu: Exploring augmented reality game design with uncertain AI-based interaction

Minji KIM

Kyungjin LEE

Rajesh Krishna BALAN
Singapore Management University, rajesh@smu.edu.sg

Youngki LEE

Citation

Bubbleu: Exploring Augmented Reality Game Design with Uncertain AI-based Interaction

Minji Kim
mjkim3035@snu.ac.kr
Seoul National University
Seoul, Korea

Kyungjin Lee
jin11542@snu.ac.kr
Seoul National University
Seoul, Korea

Rajesh Balan
rajesh@smu.edu.sg
Singapore Management University
Singapore

Youngki Lee
youngkilee@snu.ac.kr
Seoul National University
Seoul, Korea

ABSTRACT

Object detection, while being an attractive interaction method for Augmented Reality (AR), is fundamentally error-prone due to the probabilistic nature of the underlying AI models, resulting in suboptimal user experiences. In this paper, we explore the effect of three game design concepts, Ambiguity, Transparency, and Controllability, to provide better gameplay experiences in AR games that use error-prone object detection-based interaction modalities. First, we developed a base AR pet breeding game, called Bubbleu that uses object detection as a key interaction method. We then implemented three different variants, each according to the three concepts, to investigate the impact of each design concept on the overall user experience. Our user study results show that each design has its own strengths and can improve player experiences in different ways such as decreasing perceived errors (Ambiguity), explaining the system (Transparency), and enabling users to control the rate of uncertainties (Controllability).

CCS CONCEPTS

• **Applied computing → Computer games**; • **Human-centered computing → Mixed / augmented reality**; **Human computer interaction (HCI)**; • **Computing methodologies → Object detection**.

KEYWORDS

computer vision, vision sensing, Human-AI Interaction

1 INTRODUCTION

Augmented Reality (AR) games have newly adopted diverse interfaces and interaction methods [10, 12, 89]. In particular, recent advances in Artificial Intelligence (AI) and computer vision techniques have opened up sophisticated ways to integrate the real world and virtual game contexts for a high-quality player experience. An important example is object detection which allows AR games to interpret objects in the camera scene [94], enabling tangible interactions with diverse real-world objects.

Object detection techniques, however, have fundamental limitations due to their statistical nature based on Deep Neural Networks (DNN). Even the state-of-the-art DNN models trained with huge datasets inevitably suffer from unpredictable inference errors. The error rates increase when the observed scenes are beyond the scope of training data [38, 85]. Such errors during gameplay prevent players from controlling the game as they intended. In this light, understanding how errors in object detection occur and impact the gameplay experience is a critical issue when adopting object detection as a key interaction method in AR games.

In both fields of AI and Human-Computer Interaction (HCI), it has been recognized as an important research topic to build usable applications with uncertain AI solutions and enable quality user experiences. Prior studies investigate the effect of AI errors on the interaction [29, 91] and several works explore design guidelines to improve user experiences with the uncertain results [7, 9]. However, the investigations are in an early stage and focus on a few specific AI-driven applications such as chatbots [43], voice agents [66], and recommendation systems [32] while games remain as an underexplored yet important application domain. Games require unique considerations to adopt AI solutions due to the distinctive characteristics of game design [23, 39, 78].

In this paper, we aim to shed light on how to utilize uncertain AI solutions as game interaction mechanics for game designers and HCI researchers. In particular, we deeply investigate how to overcome the inherent uncertainties of DNN-based object detection with interaction designs in the context of AR games. Our work is one of the first attempts to provide useful insights into designing game interactions with fast-penetrating AI solutions. Such findings will especially be important for emerging AR games where AI solutions are essential for physical scene analysis and seamless virtual content rendering. In addition, our work extends existing HAI studies with a new target application and findings.
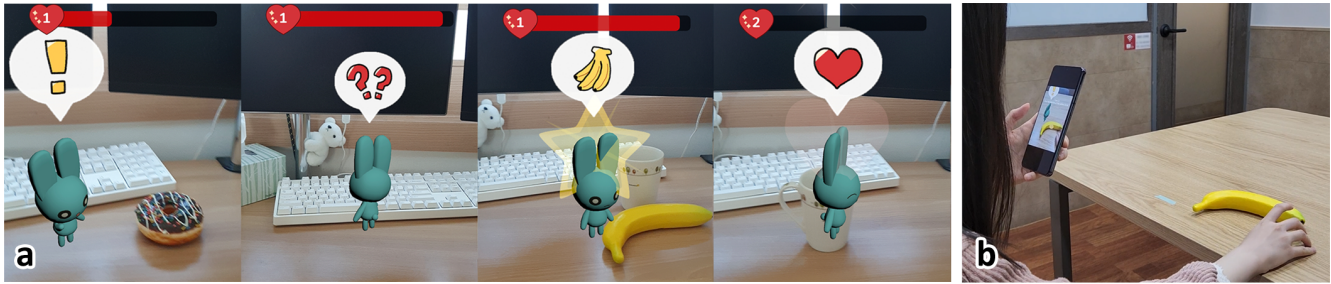
Figure 1: Gameplay of Bubbleu. a) Screenshots of the gameplay. b) A player playing Bubbleu using a real object.

Emerging AR game applications with new AI-based interaction modalities have a large design space yet to be explored. We conduct a sequence of three essential studies to dissect the design space and rigorously study the effects of various designs (see Section 3 for study procedure overview). In our first study, we design Bubbleu, an AR pet breeding simulation game for handheld mobile devices (Figure 1). Bubbleu utilizes object detection as the sole interaction method, allowing players to control the game by manipulating physical objects in the camera scene. In our second study, we improve Bubbleu with three design concepts - *Ambiguity*, *Transparency*, and *Controllability* - to preserve the gameplay experience with the presence of different types of object detection errors (see Section 5.3 for details). Through the preliminary gameplay of Bubbleu, we learn that various object detection errors have different negative effects on the intended player experience. We carefully choose the three designs based on prior HAI and game design guidelines to mitigate such negative impact.

We finally conduct a user study with 36 participants to evaluate the effectiveness of the three Bubbleu variants, in improving the player experience upon object detection errors. Through the experiments, we aim to answer the following two research questions:

RQ1. *How do errors occurring in the object detection interaction affect the user experience in AR games?*

RQ2. *What possible game designs can improve the player experience when object detection errors occur in the gameplay?*

The results show that each design variant has different effects on preserving user experience by decreasing the perception of errors (*Ambiguity*), making users understand the ambiguous interaction (*Transparency*), and enabling the users to control the uncertainties (*Controllability*). *Ambiguity* design decreased the success rate of the interaction for 19.77 % in the erroneous environment. *Transparency* and *Controllability* design did not result in a clear decrease in error rates but had high user preference compared to the baseline with their abilities to explain and control the uncertainties.

The main contributions of this paper are as follows:

- We systematically characterize the types of object detection errors that lead to negative gameplay experiences.
- We show the potential of object detection as an interaction method for AR games by implementing a novel AR game called Bubbleu that uses object detection as a key interaction modality.
- We investigate how different game designs from three concepts – *Ambiguity*, *Transparency* and *Controllability* – can

possibly reduce the negative impact of object detection errors and improve the Bubbleu gameplay experience.
- We present design implications for future games to adopt object detection-based game interaction.

2 RELATED WORK

2.1 Object Detection-based Applications

Object detection is a long-studied topic in computer vision. It enables seamless interaction with real-world objects in the digital context with two useful features: estimating the locations and categories of various objects [94, 96]. The synergy between large-scale data [21, 56] and breakthroughs in DNNs has shown unprecedented in-the-wild performance; various objects in a scene can be detected and recognized with a single image input [13, 48, 60, 79]. This technology paved the way for numerous applications including autonomous driving [80] or Amazon's in-store autonomous checkout [82]. In these applications, even a single failure can lead to fatal results (e.g., accidents with autonomous driving cars [35, 63]), and thus they use many high-end cameras and powerful computing equipment to minimize the uncertainty.

A large set of applications were proposed that use lightweight object detection technology for the automation of everyday tasks. Some examples include optical character recognition (OCR) for image-to-text [77], face detection for camera auto-focus [4, 65], pet monitoring [33], and hands-free control of IoT devices with smart cameras [40, 73, 86]. Many studies also showed that object detection enables efficient video analysis for note-taking [16], caption generation [49], and sports analysis [20]. Unlike mission-critical applications, these applications achieved sufficient accuracy to minimize human labor and provided intuitive interfaces to overcome uncertainties with explicit user feedback.

Recently, a large body of work has explored the design space of integrating AR and object detection-based interactions. In particular, many applications were proposed for educational purposes that used context awareness and gamification. Draxler et al. [25] presented a context-based grammar learning application, which recommends a sentence related to the types and positions of the objects in the scene. Kang et al. [42] presented ARMath, a math education application with an object detection-based tangible object manipulation interaction. Kwon et al. [50] proposed one of the first systems that support language guidance during parent-child

interactions. Other works aim to provide new entertaining experiences integrating virtual content and real-world objects. Liang et al. [55] implemented a virtual reality pet system that generates context-aware behavior of a pet related to the real-world scene. Putze et al. [70] presented an AR guessing game that utilizes object detection. These works all focus on showing the potential of novel interactions, however, the consequences of object detection uncertainty remain underexplored.

2.2 Uncertainty in Human-AI Interactions

The uncertainty of AI-based systems has been extensively studied in the field of Human-AI Interaction (HAI), with various terms such as eXplainable AI (XAI), transparent AI, and responsible AI being coined [5]. A large body of work studied how users create mental models when encountering uncertain and unexpected AI behaviors. Studies have shown that the timing of encountering AI model errors [44, 68], level of expertise [53, 67], and fidelity of the information used to explain the AI model results [45, 93] affects the user's trust and understanding towards the system. Another set of studies explored how to guide users to overcome AI model errors. Das et al. proposed new ways to provide non-experts with explanations on how to overcome AI-based robot failures [18]. Lindvall et al. explored the design space of interacting with imperfect AI (e.g., hiding underlying probabilities) for digital pathologists [57]. Kocielnik et al. conducted a case study with a scheduling assistant AI application to understand user expectations towards imperfect AI [47]. They found that higher recall resulted in a higher perception of accuracy compared to higher precision. In addition, giving control to the user to adjust the precision and recall trade-off resulted in higher user acceptance. In this work, we deeply investigate AI uncertainty in the important but under-explored context of AR games where the findings can differ from prior work even when applying similar design strategies.

Prior works have attempted to generate a set of guidelines for UI/UX practitioners to handle uncertainties. Yang et al. claimed that the unique challenges of designing human-AI interactions come from the *capability of uncertainty - uncertainties surrounding what the system can do and how well it performs* [91]. Amershi et al. presented a set of general guidelines applicable to AI applications [7] including the guidelines to handle the cases when AI is wrong. Furthermore, companies are taking initiatives to ensure responsibility and accountability for commercialized products. Google's People+AI research group (PAIR) presented a guidebook [3] that highlights things to consider for error handling. Apple [2] and IBM [1] also proposed a set of guidelines to serve similar purposes.

In our work, we aim to discover a set of new design implications that account for game-specific contexts. Our findings may align with or contradict existing knowledge as many of them are performance-centric and miss the notion of AI as play. Furthermore, we focus on games where AI errors impact the player experience directly rather than games where AI-controlled characters indirectly impact the experience.

2.3 Uncertainty in Games

Errors during gameplay have become diversified, with the increasing complexity of games and the adoption of new technology. Prior

work has studied the influence of various error types on the payer experience. For instance, several studies explored the impact of network latency in various types of games [11, 36, 37, 59]. Others studied errors regarding input accuracy when adopting novel interaction modalities [19, 74]. Many commercial games denote any type of error as bugs and something to avoid [54, 81] since they can violate the player's intentions and cause frustration. Indeed, users often perceive most of the conventional errors as system failures or malfunctions [3].

As no error-free AI systems exist, prior work has explored new game designs to make AI-based games error-tolerable. Zhu et al. conducted a systematic review on AI-infused games [95]. One study showed that players could be generous about bugs as long as they were entertained [81]. Other studies suggested using different input modalities or controlling granularity [61]. ARMath provided a fixation UI that allowed users to fix the AI results using gamification features [42]. Zargham et al. proposed anticipatory error handling, allowing the game to choose the best action for achieving the goal when the user input is not accurately recognized [92]. In line with these works, we explore various designs to handle AI errors that consider the characteristics of the game and player expectations.

3 STUDY OVERVIEW

We illustrate our overall study procedure in Figure 2(a) along with a detailed description of individual study in Figure 2(b). We obtained IRB approval for all of our user studies. Also, the study process followed the government's COVID-19 health protocol. In this section, we overview the three studies while we explain further details of the methodology and study designs in the corresponding sections.

The first study is a pre-design workshop with 7 participant group of HCI researchers and game design experts to understand the design space of objection detection-based games (Study 1 in Figure 2). Through a two-phase workshop and a post-workshop survey, we i) explored how object detection can be used as a game interaction modality, ii) investigated how object detection errors can affect the gameplay, and iii) constructed a game design suitable for understanding the effects of the errors. Based on the results of the pre-design study, we implemented the baseline game Bubbleu. In the second study, we conducted another design workshop to explore how the baseline design can be improved to minimize the impact of object detection errors on the player experience (Study 2). As a result, we converged design improvement ideas into three different concepts - *Ambiguity*, *Transparency*, and *Controllability*. We implemented three variants of Bubbleu following each design concept. Finally, we conducted a user study of 36 participants including two types of experiments and interviews (Study 3). We aimed to investigate i) if the three design concepts reduce the actual and perceived errors during gameplay and ii) how they affect the player experience, both in natural gameplay environments and controlled environments with manually injected errors.

4 STUDY1: PRE-DESIGN EXPLORATION

4.1 Methodology

We conducted a two-phase design workshop to explore object detection-based AR games. We recruited a group of researchers including two game design experts with more than four years of
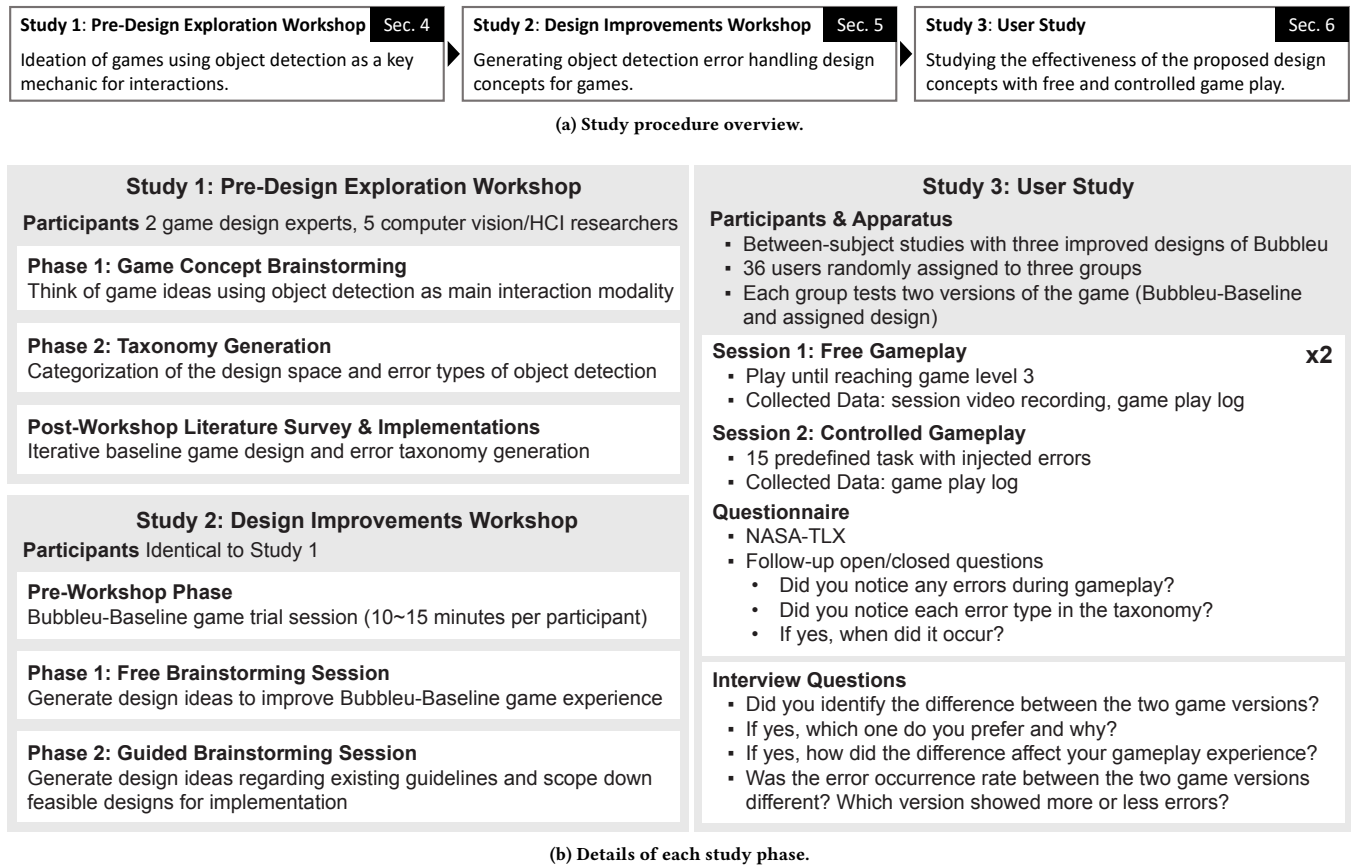
Study 1: Pre-Design Exploration Workshop Sec. 4	**Study 2**: Design Improvements Workshop Sec. 5	**Study 3**: User Study Sec. 6
Ideation of games using object detection as a key mechanic for interactions.	Generating object detection error handling design concepts for games.	Studying the effectiveness of the proposed design concepts with free and controlled game play.

(a) Study procedure overview.

Study 1: Pre-Design Exploration Workshop

Participants 2 game design experts, 5 computer vision/HCI researchers

Phase 1: Game Concept Brainstorming
Think of game ideas using object detection as main interaction modality

Phase 2: Taxonomy Generation
Categorization of the design space and error types of object detection

Post-Workshop Literature Survey & Implementations
Iterative baseline game design and error taxonomy generation

Study 2: Design Improvements Workshop

Participants Identical to Study 1

Pre-Workshop Phase
Bubbleu-Baseline game trial session (10~15 minutes per participant)

Phase 1: Free Brainstorming Session
Generate design ideas to improve Bubbleu-Baseline game experience

Phase 2: Guided Brainstorming Session
Generate design ideas regarding existing guidelines and scope down feasible designs for implementation

Study 3: User Study

Participants & Apparatus
- Between-subject studies with three improved designs of Bubbleu
- 36 users randomly assigned to three groups
- Each group tests two versions of the game (Bubbleu-Baseline and assigned design)

Session 1: Free Gameplay x2
- Play until reaching game level 3
- Collected Data: session video recording, game play log

Session 2: Controlled Gameplay
- 15 predefined task with injected errors
- Collected Data: game play log

Questionnaire
- NASA-TLX
- Follow-up open/closed questions
  - Did you notice any errors during gameplay?
  - Did you notice each error type in the taxonomy?
  - If yes, when did it occur?

Interview Questions
- Did you identify the difference between the two game versions?
- If yes, which one do you prefer and why?
- If yes, how did the difference affect your gameplay experience?
- Was the error occurrence rate between the two game versions different? Which version showed more or less errors?

(b) Details of each study phase.

Figure 2: Overview of our study methodology.

experience and five researchers who have more than two years of experience in HCI and computer vision (4 females and 3 males aged between 24~28). We recruited the participants considering that the study is at the intersection of computer vision, HCI, and game design. The recruitment was done through our intranet channel and the participation was voluntary. All the participants had hands-on experience with training and testing state-of-the-art object detection models. The workshop was held at an on-site meeting room following the government's COVID-19 health protocol, and lasted two hours in total.

In the first phase of the workshop, the participants were asked to generate game concept ideas using object detection as the primary interaction modality. We provided the MDA framework [39], a formal approach for game design, to facilitate the ideation process for non-experts in game design. The participants were guided to generate new ideas regarding the three main components of games; i) the rule and system of the game (**M**echanics), ii) the run-time behavior of the game (**D**ynamics), and iii) the desirable emotional responses evoked by the game (**A**esthetics). As we aimed to explore the design space of object detection-based interaction, we provided more detailed information on the mechanics [26, 76] and encouraged the participants to generate ideas focusing on the mechanics utilizing object detection. Experienced game designers preferred not to be limited by the framework. This phase of the study lasted
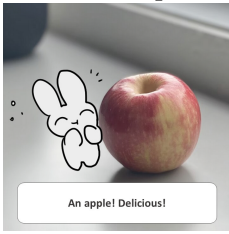
approximately one hour. As a result, we converged to 33 interesting game ideas, including *'a room escape game to interact with real-world objects (e.g., a real key to open a door)'*, *'a shooting game to shoot real-world objects around them'* and *'a game to create a bouquet by collecting real flowers'*.

In the second phase, we collectively discussed the various suggestions to organize our findings. We conducted an affinity diagramming exercise to categorize the design space of games using object detection as a primary interaction modality. Table 1 shows the categorization along with examples. Then, we extensively discussed and chose a single prototype game with a rich set of interactions to highlight the potential problems caused by object detection errors. We also discovered that different types of object detection errors have varying impacts on gameplay. Thus, we conducted a thorough literature review after the workshop to identify the errors found in prior studies [15, 38] and generated a taxonomy. We organize the findings of the workshop into two sections; i) design space of object detection-based games and our design choices (Section 4.2), and ii) taxonomy of object detection error types (Section 4.3).

4.2 AR Game Design with Object Detection

Table 1 presents the final four designs categories and example games, separated by two criteria: i) whether they require interaction with multiple objects at the same time, and ii) whether the user

Table 1: Categories of AR game designs that utilize object detection as a key mechanic. A) A quiz game where a virtual character asks the player to show an object to solve a short quiz. B) A pet breeding game where the player can feed a virtual pet by manipulating a real food object. C) A stroll game where a virtual fairy talks about the scene while walking outdoors. D) An education game where the player should manipulate real objects to solve a math problem.

	Static Manipulation	Dynamic Manipulation
	A. Quiz game	**B. Virtual pet**
Single Object	What being has four legs, then two, and then three?	An apple! Delicious!
	C. Stroll assistant	**D. Math education**
Multiple Objects	You're in the city!	How many apples are in the basket?

Table 2: Taxonomy of object detection errors.

Error Type	Description and Metric
Missing detection	Fails to localize and recognize the object in the scene (False negative) $AreaOf(P_{bbox}) = 0, AreaOf(GT_{bbox}) > 0$
False detection	Localize and recognize a non-existing object (False positive) $AreaOf(P_{bbox}) > 0, IoU(P_{bbox}, GT_{bbox}) < t_b$
Mislocalization	Correct recognition of an object in the scene but in the wrong location $t_b < IoU(P_{bbox}, GT_{bbox}) <= t_f$
Misclassification	Correct localization of object in the scene but recognized as a different class $ClassOf(P_{bbox}) \neq ClassOf(GT_{bbox})$

*$AreaOf$ is the size of the bounding box, IoU is the intersection-over-union quantifying the overlapping region of two boxes (threshold values are empirically determined, e.g., $t_b = 0.1$, $t_f = 0.5$), $ClassOf$ is the classified object type.

dynamically manipulates the object (e.g., moving it around) during the interaction - versus just selecting the object. We also considered the third aspect, i.e., the types of objects to be detected and used in the game. However, as the accuracy of object detection with diverse object types is heavily affected by the specific DNN models and training data used rather than the game design, we excluded this third factor in our final design consideration.

Design choices of our prototype game. From the workshop results, we carefully designed a prototype game that could clearly show the impact of object detection errors. In particular, we picked the category of single-object and dynamic manipulation. Firstly, the single-object manipulation highlights the type of object detection errors and their effects. We excluded multi-object interactions since different types of errors could be blended, making the analysis non-trivial. In addition, we chose dynamic manipulation over static one since it not only enables a fun game experience but also generates diverse types of errors. With these considerations, we built a virtual pet breeding game named *Bubbleu* (see Section 5).

4.3 Errors in Object Detection Interaction

Table 2 is a generated taxonomy of errors caused by deep learning-based object detection during gameplay. The taxonomy includes four error types – *Missing detection*, *False detection*, *Mislocalization* and *Misclassification* – regarding the two capabilities of object detection: i) detecting the position of an object and ii) classifying

its category. Note that even though these errors can be alleviated with improved object detection algorithms, they cannot be fully eliminated due to the probabilistic nature of deep learning-based approaches [69] and mismatches between the training data and highly diverse gameplay scenarios. We observed that the errors indeed occur in Bubbleu gameplay sessions at different frequencies (Section 6.2.1) and discussed the potential causes of those errors (Section 7.3).

5 STUDY2: BUBBLEU DESIGN

5.1 Methodology

Based on the findings of the pre-design workshop, we built a baseline prototype of Bubbleu that utilizes object detection as the primary interaction modality. To enrich our observations, we iteratively designed Bubbleu's interaction modalities to expose the effects of the different error types listed in Section 4.3. The final design of Bubbleu is represented by the Finite State Machine (FSM) architecture shown in Figure 4. We explain how the different error types are reflected in the Bubbleu design (using the FSM architecture) in Section 5.2. This baseline Bubbleu (referred to as *Bubbleu-Baseline*, hereinafter) was used to study the effects of different types of object detection errors during gameplay.

We then conducted a second design workshop in two phases with the same group of researchers who attended the first workshop. Before the start of the workshop, each participant played a trial session with *Bubbleu-Baseline* to understand the game features and to experience the impact of object detection errors during gameplay. In the first phase (for half an hour), the participants engaged in a free brainstorming session where they were encouraged to suggest ideas for improving *Bubbleu-Baseline*. Many of these ideas included adding or completely changing the features of the game (e.g., making the player teach the rabbit when errors occur, adding another pet to distract the user from errors, asking the players to

Table 3: Types of interactable objects and the consecutive interactions in Bubbleu. The types of objects are selected from COCO dataset labels.

Name	Behavior of the Pet	Interactable Objects
Eat	*The pet eats the object.*	Apple, Banana, Orange, Donut, Carrot, Broccoli
Wash	*The pet washes its face.*	Cup, Bottle
Play	*The pet follows the hand.*	Hand

move objects only when no interaction has occurred within a short time period, etc.).

For the second phase (also for half an hour), we organized a guided brainstorming session to narrow down the design space. We first clarified the FSM architecture of Bubbleu and showed the types of errors that could occur in each state. Then, we provided a set of design guidelines that could help stimulate additional ideas. We carefully selected the guidelines from existing game design theories [23, 39, 83, 90] and Human-AI interaction guidelines [7, 51]. The participants were encouraged to generate new ideas or refine their existing ideas.

We collated all the ideas generated from the two phases and conducted an affinity diagramming exercise to classify and converge the ideas into three different concepts (explained in Section 5.3). Then, we chose a set of specific ideas to implement. After the workshop, we performed several iterative prototyping rounds by testing and improving the implemented designs. Section 5.3 describes how we extracted the three design concepts and the final chosen designs. Note that the selected designs from the workshop are not exhaustive or representative. We discuss some of the limitations and possible improvements in Section 7. Figure 5 shows how the workshop was conducted and provides the full list of converged ideas in Appendix 10.

5.2 Baseline Game Design of Bubbleu

We first describe how the baseline version of Bubbleu works.

5.2.1 Overall Gameplay. Bubbleu is an augmented-reality pet breeding simulation game played on a handheld smartphone. The game is played by manipulating real-world objects on flat surfaces (e.g., floors or desks) captured by the device's rear camera. As the game begins, a virtual rabbit pet appears on the surface. The goal is to breed the virtual pet seen on the screen by fulfilling its needs and gaining EXP (EXperience Points) to level up the pet. The pet has three needs; eat, wash, and play. The player can fulfill the pet's needs by bringing real-world objects near the pet. For example, the player could place an apple in front of the virtual pet. When an apple is detected in the scene, the pet will eat it, and its satisfaction gauge for that need will increase. As the satisfaction gauges are fulfilled, the player is rewarded with EXP scores, which in turn increases the level of the pet.

The pet can detect and interact with 9 different types of objects (Table 3). In particular, the player can feed the pet with food objects and wash the pet with the *cup* and *bottle* objects. To move the pet to specific positions, the player needs to show their hand on the
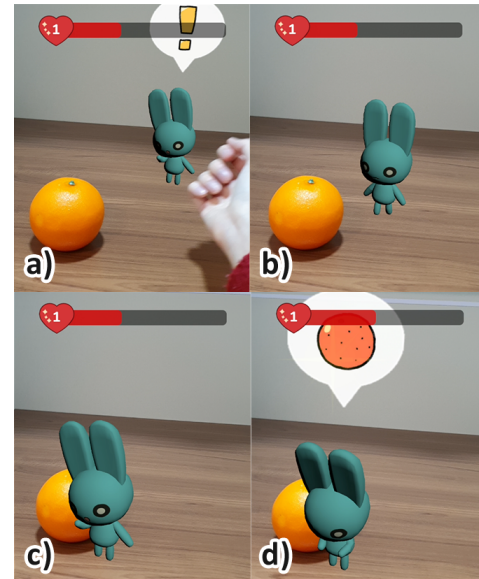

Figure 3: The gameplay state sequence of Bubbleu. a) The pet becomes curious after detecting an interactable object. b) The pet moves to the position of the detected object. c) The pet checks if the detected object is still in the same position. d) The pet interacts with the detected object.
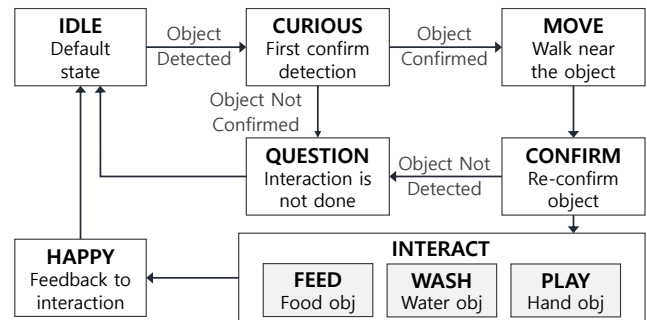

Figure 4: The Finite State Machine (FSM) for Bubbleu.

screen – this will cause the pet to follow the hand. The pet's need to play is fulfilled when the player consecutively leads the pet to follow his hand 5 times.

5.2.2 Behavior of the Pet. The behavior of the pet follows a 7-state FSM as illustrated in Figure 4. Figure 3 describes an example transition sequence for major states. The initial and default state of the game is IDLE, where the virtual pet stands at a fixed location. When the system detects an interactable object in the camera input, the pet switches into the CURIOUS state. In this state, the pet indicates its curiosity while the system identifies the class of detected objects (Figure 3(a)). If the object is detected and recognized for 10 consecutive frames, the pet successfully transitions into the MOVE state and walks toward the detected position (Figure 3(b)). After the pet moves close to the detected object, the CONFIRM state is

Figure 5: Conducting game design workshop for Bubbleu improvements.

executed and checks if the detected object is still in the right position for another 10 consecutive frames (Figure 3(c)). A successful confirmation causes a transition to the INTERACT phase, where the virtual pet performs different interactions depending on the class of the detected object (Figure 3(d)). If either the CURIOUS or CONFIRM fails, the pet transitions to a QUESTION state that shows an animation of the pet being confused. After finishing the INTERACT or the QUESTION action, the pet transitions back to the IDLE state and waits until another object is detected.

5.2.3 Object Detection Components of Bubbleu. We adopted SSD (Single Shot Multibox Detector) Lite [60] model trained by the COCO (Common Objects in Context) object detection dataset [56] to perform object detection in Bubbleu. We chose the model as it was optimized for mobile devices (did not result in any noticeable lag), and the error rate was only a little higher (COCO mAP@.5=46.5 %) than the state-of-art model (COCO mAP@.5=56.8 % [84]). Thus, the findings of this paper will still apply to other object detection models. Bubbleu confirms the detection only after checking that the object is consistently detected with a confidence level higher than the empirically chosen threshold ($th >= 0.55$) for 10 consecutive frames in the CURIOUS and CONFIRM states.

We observed that different types of object detection errors occur at various states. First, missing detection and false detection influence the IDLE and CHECK states. For instance, in the IDLE state, a missing detection does not trigger the transition to the CURIOUS state, while a false detection makes an unintended transition. In the CHECK state, a missing and false detection incurs a wrong state transition to the QUESTION and INTERACT states, respectively. Mislocalization errors appear between CURIOUS and MOVE states, affecting all subsequent transitions. When it occurs, the virtual rabbit moves to the wrong position and performs interactions there (e.g., the rabbit eating an apple far away from it). Lastly, misclassification errors occur in the INTERACTION state. They lead to activating wrong interactions (e.g., the washing action when the detected object is an apple). To accommodate these errors, we carefully design *Bubbleu-Baseline* to handle these object detection errors at different FSM states.

5.3 Error Handling Designs

From the second workshop, we classified the design ideas into three concepts - Ambiguity, Transparency, and Controllability. In this

section, we define these concepts and detail how these concepts were implemented into specific Bubbleu variants for the user study.
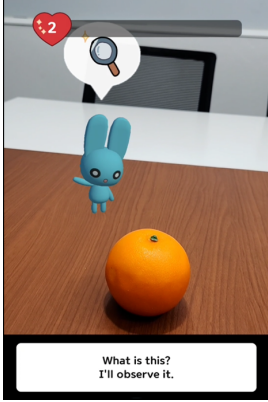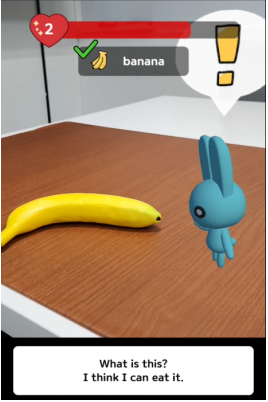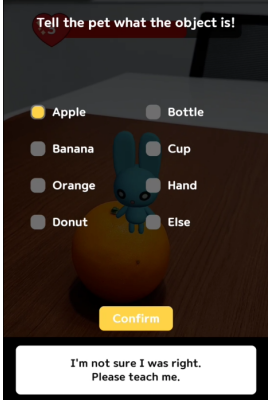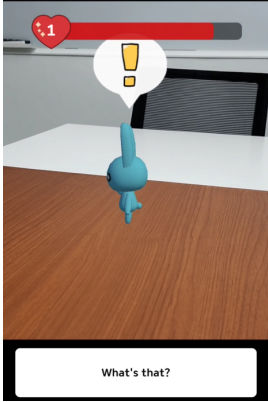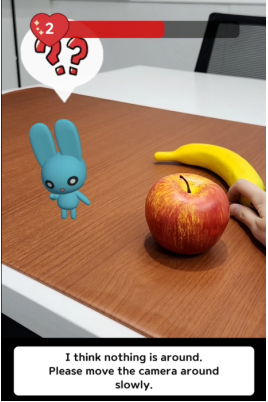
5.3.1 Ambiguity. Ambiguity is a design concept to overcome negative experiences caused by technical limitations by pursuing multiple interpretations of the expression [14, 28, 75]. This design concept can be applied to Bubbleu by abstracting the interaction feedback into ambiguous expressions to prevent object detection errors from being perceived by the players. For example, classifying an orange as an apple is a common error in Bubbleu. When the system displays an exact icon of the object (e.g., displaying the "apple" icon when it detects an *apple*), the player easily discovers the error. However, when the detection results of *apple* and *orange* are abstracted and displayed as the same "eat" icon, the player does not notice whether an error has occurred. The *Missing detection* errors (e.g., the pet does not react when an object is placed in front of the pet) could also leverage ambiguity by having the pet do random interactions when no object is detected for a certain period. We incorporated various ideas related to ambiguity to create a new version, *Bubbleu-Ambiguity*.

Implementation of *Bubbleu-Ambiguity* We define a concept of *level-of-interaction*, which divides interactions into three levels by confidence scores. When an object is detected with a low confidence score, the pet in *Bubbleu-Ambiguity* performs a new type of interaction called *observe*. The *observe* interaction is an abstract interaction that can be done with any type of object detected in Bubbleu. Unlike *eat* and *wash*, the pet doesn't move near the object or display the object icon. Instead, it only looks ("observes") at the object. This interaction aims to hide *Mislocalization* and *Misclassification* errors while still allowing the player to interact with the object. *Bubbleu-Ambiguity* divides *eat* and *wash* interactions into three levels. When the confidence level is *low*, the pet *observes* the object instead of eating or washing (A-1 in Table 4). When the confidence is *medium*, the pet displays the higher group of the object type. For example, when an apple is detected with *medium* confidence, the pet shows the *eat* gesture and displays the *eat* icon. In contrast, when the same object is detected with *high* confidence, the pet shows the *eat* gesture along with the *apple* icon and informs that it is eating an apple in a dialogue. To reinforce that *observe* belongs to a successful interaction, the players receive EXP as a reward of *observe*.

The *observe* interaction is also played randomly in the IDLE state (A-2 in Table 4). When this triggers, the pet will start looking at a random position, and an observing gesture will be displayed. This feature attempts to hide *Missing detection* errors by showing natural feedback when the system fails to detect an object for a long time.

5.3.2 Transparency. Transparency has long been considered an indispensable principle for designing understandable systems [27, 30]. Transparent systems help users to understand the behavior of complicated sub-systems such as algorithms [8, 24, 46, 52] or AI models [58, 64, 72]. In particular, transparency helps improve user acceptance, trust, fairness, and usability [64]. During the workshop, multiple participants suggested different designs for providing transparent information about the detection results. For example, one participant suggested that the pet should first identify how confident the pet is about the detection before the interaction. Another idea was that the pet should provide additional information on

Table 4: Applications of three design variants in error-expected situations.

Error Situations	Ambiguity	Transparency	Controllability
Interaction proceeds with low confidence *i.e., False detection, Misclassification, or Mislocalization is expected*	**A-1. Level-of-interaction** The pet shows ambiguous interaction depending on the current confidence score.	**B-1. Detection report** Displays detected object and confidence score during full detection process.	**C-1. Ask for correction** After the interaction, the pet sometimes asks the player to correct the detection result.
No interaction happens for a long time period *i.e., Missing detection is expected*	**A-2. Random observe** The pet randomly *observes* the location where no object is detected at	**B-2. Guide to recalibrate** The pet guides the player to recalibrate the scene for better detection.	**C-2. Threshold control** The player can use a hand gesture to lower the confidence score threshold.

why the failures happened and how to overcome them. We implement *Bubbleu-Transparency* to investigate whether providing transparency results in positive effects when unexpected errors occur.

Implementation of *Bubbleu-Transparency* *Bubbleu-Transparency* displays the detected object class and the confidence level during the full detection process (B-1 in Table 4). The interaction reward changes depending on the detected confidence level. When the confidence level is *low*, the pet would ask the player to try detecting the object again instead of performing an interaction – here, no EXP will be provided. When the confidence level is *high*, an additional effect would be played and the EXP reward will be slightly higher than common. Note that, unlike the other two designs, this

behavior is always shown regardless of the confidence score to maintain consistency of the information.

When no object is detected for a long time period, the pet tells that no object is being detected and provides tips on how to improve the object detection accuracy (B-2 in Table 4). For example, the pet will sometimes say *"Please move the camera around slowly"* when no object is detected for a while.

5.3.3 Controllability. Controllability is an essential design element for digital games [22, 39, 88]. In HAI, controllability is used to enhance the trust of users by explaining what is happening and how they can control the system to react appropriately [6, 62]. In Bubbleu, we adopted controllability to allow the user to tune the

Table 5: Participant distribution in three groups of the user study. All participants were in the age group of 20-39.

Group	Participant	Gender (M/F)	AR Experience Level (1-4)
Ambiguity	P1-P12	7/5	2.08 ± 0.80
Transparency	P13-P24	4/8	2.17 ± 0.71
Controllability	P25-P36	6/6	2.50 ± 0.90

capability of the object detection models. Some suggested ideas for controllability were to let users control the sensitivity of the object detection by, for example, allowing users to force the pet behavior in a dark environment. Other suggestions were to explicitly ask for corrections to improve the object detection models. We thus implement *Bubbleu-Controllability* to understand whether providing controllability improves the player experience.

Implementation of *Bubbleu-Controllability* The final variant, *Bubbleu-Controllability*, provides a feature that lets players control the confidence threshold of the detection. When no object is detected for a while, the pet will ask the player to show their hand on the screen (e.g., the message *"I don't detect anything. Please show me your hand if there is an object to detect."* will appear). If the player shows their hand, the confidence threshold lowers so that the sensitivity of the object detection would be increased (C-2 in Table 4). This allows the players to avoid *Missing detection* errors, while chances of *False detection* errors also increase.

Bubbleu-Controllability also asks players to correct the detection results (C-1 in Table 4). When the detection confidence level is *low*, the pet questions the player about whether the previous object detection was performed correctly. This design thus allows users to provide feedback to the object detection subsystem about its runtime performance. The given feedback can be utilized to dynamically improve the system to increase performance.

5.3.4 Applying the three concepts to Bubbleu. Each design concept is applied in the following way. Firstly, Bubbleu estimates the two potential situations where errors likely occur: i) situations where no interaction has happened over a long period (i.e., *Missing detection* errors are expected), and ii) situations where interactions are occurring but the confidence score is lower than the pre-determined threshold (0.65, decided empirically) (i.e., *False detection*, *Misclassification*, or *Mislocalization* errors are expected). In these two situations, each variant behaves differently following its own implementation (Table 4). Note that the game system cannot predict whether the error has truly occurred. These behaviors could also occur in non-erroneous circumstances.

6 STUDY3: USER STUDY

6.1 Study Design

6.1.1 Participants and apparatus. We recruited 36 participants in South Korea via an online survey. The participant group consisted of 17 males and 19 females between 20 to 39 years of age. All of them had a level of education above a bachelor's degree. During recruitment, we asked each participant to report their prior

experience level in AR applications in a 4-point Likert survey; 5 participants have never used AR applications, 20 have used but not consistently, 8 have used consistently, and 3 have developed or researched AR applications. To reduce selection bias, we divided them randomly into three groups of 12 participants except for the 5 complete novices and the 3 expert developers who were spread across the groups. Table 5 shows the distribution of the participants. We conducted between-subjects experiments with the three improved designs of Bubbleu. Each group was assigned to play one of the variants (*Bubbleu-Ambiguity*, *Bubbleu-Transparency*, or *Bubbleu-Controllability*) along with *Bubbleu-Baseline*, which serves as a common baseline for the within-subjects comparative analysis. The results were analyzed within each group only with no comparison between the groups.

The study was conducted on-site in a conference room of our university building. At the start of the study, each participant was provided with game instructions and informed about the study procedure. Then, they signed a document that they agreed to participate in the experiment. Before starting the game, the participants went through a short tutorial gameplay session.

For each version of the game, the participants went through two sessions of experiments: free gameplay and controlled gameplay. In the free gameplay session, we asked participants to play the game naturally without instructions. In the controlled gameplay session, the participants were guided to perform specific interaction tasks. Here, object detection errors were manually injected into some of the tasks. The order between the assigned variant of Bubbleu and the baseline Bubbleu was randomly counterbalanced. Every session was conducted in a think-aloud manner. One researcher resided in the same room throughout the entire study session to assist the participants with the experiment procedure when requested. The entire user study lasted 1 to 1.5 hours per participant and was carried out in a span of three weeks. During all sessions, the game screens and participants' behaviors were video-recorded under consent.

- **Free Gameplay Setup.** The participants could control the smartphone device and manipulate the objects freely. Each participant played the game until they achieve *level 3* of the game, which requires about 15 successful interactions. Six interactable objects were provided: apple, banana, orange, donut, bottle, and cup. We asked them to interact with the pet using every object at least twice.
- **Controlled Gameplay Setup.** We adopted a Wizard-of-Oz approach for the controlled gameplay session. The participants were guided to follow 15 predefined tasks given through the game message. We randomly divided the 15 tasks into 7 successful tasks and 8 erroneous tasks. For the successful tasks, object detection was performed at 100 % accuracy. On the other hand, for the erroneous tasks, we injected one of the 4 types of errors (listed in Table 2). Each error type occurred twice. The participants were asked to perform three actions in sequence with a target object: i) moving the target object to the indicated position on the desk, ii) observing the interaction between the pet and the object, and iii) removing the object from the desk. Upon the completion of each task, the participant was instructed to

press either the *"Report"* or *"Continue"* buttons. If the participant pressed the *"Report"*, they could then select the type of error they have noticed in the popup UI. Otherwise, they would proceed directly to the next task if they pressed *"Continue"*. The participants were only allowed to manipulate the single object specified in the instructions during the task. We used the same six interactable objects from the free gameplay study.

After each gameplay session, we asked the participants to first fill out the questions from NASA Task Load Index (TLX) questionnaire [34] in 7-point Likert scales to validate whether the improved design caused an additional mental workload. Then, the participants were given closed and open questions in a semi-structured interview format. The essence of the questions was whether the participant noticed any errors and if the types of errors could be identified. At the end of the two sessions, we conducted an additional interview to ask the participants about the overall game experiences and differences between the baseline and improved design. We asked if the participants noticed the difference between the two game versions, which one they preferred, and which one showed fewer errors. The specific questions for both interviews are shown in Figure 2. All the interviews were audio recorded under consent.

All the studies were conducted with Samsung Galaxy S21 devices. We found that the device's thermal issues could affect the gameplay by causing noticeable object detection latency. Therefore, we used two devices alternatively and attached them to a device cooler in between the sessions.

6.1.2 Data, metrics, and analysis methods. Our user study generated four sets of data for analysis.

- **Video recordings and gameplay logs (free gameplays).** We aligned the video recordings with the gameplay log and identified the following quantitative metrics for analysis: i) the total number of interaction attempts (the participants' action of placing a new object and/or moving the camera towards an object), ii) the success rates (number of successful attempts over total interaction attempts), and iii) the number and types of errors that occurred (as determined by in the taxonomy in Table 2). We used the paired two-tailed t-test to analyze the statistical significance between the baseline and a variant of improved Bubbleu. The results are reported in Section 6.2.1. Note we removed the outlier data (P3 from *Bubbleu-Ambiguity*, P16 from *Bubbleu-Transparency*, and P29 from *Bubbleu-Controllability*). For these outlier participants, their number of interaction attempts was significantly higher than the average (more than two standard deviations away) indicating that they had not learned how to play the game properly.
- **Gameplay logs (controlled gameplays).** We obtained the perceived error rate (defined as the percentage of the reported errors) from the error report logs of the controlled gameplays. We report the relevant results in Section 6.2.2.
- **Quantitative perceived task load (free and controlled gameplays).** Participants answered six survey questions

from the NASA-TLX questionnaire (regarding task load index) [34] after playing each gameplay session. We compared the scores of each question (between the Bubbleu baseline and our design variant) with the paired two-tailed t-test. The results are reported in Section 6.2.3.
- **Qualitative interview.** We manually transcribed all the interview data from the audio recordings. We used the data to analyze the users' preference towards each design variation (Section 6.2.4). We also organized our observations on the user behaviors and their answers to our questions related to error perception (Section 6.2.5). The interviews were conducted in Korean. Thus, all the reported quotes in this paper were translated by a bilingual speaker and verified by the authors.

6.2 Results

6.2.1 Success rate of interactions in free gameplay. The success rate of the interaction attempts during free gameplay is shown in Table 6. The results include the total number of attempts, the number of successful attempts, the success rate, along with the number of error occurrences (total and per type). Overall, all three designs helped improve successful interactions (with statistical significance for *Bubbleu-Ambiguity* and *Bubbleu-Transparency*).

In *Bubbleu-Ambiguity*, the results show a clear increase in the success rate, defined as the number of successes divided by the total attempts - there was a statistically significant ≈20% increase from 58.19% to 77.96%. Specifically, the *Missing detection* error decreased from 3.78 to 2.33. In *Bubbleu-Transparency*, the success rate also increased by 8.51% (from 72.18% of the baseline to 80.69%). *Bubbleu-Transparency* is the only variant that decreases the *False detection* error rate. *Bubbleu-Controllability* also resulted in a higher success rate and lower failure rate, but the numbers are statistically insignificant.

6.2.2 Perceived errors in controlled gameplay. Next, we show the perceived error rate results measured from the controlled gameplay session. Table 7 describes the percentage of error reports for 8 out of the 15 tasks with injected errors. A lower error rate means that the number of reported errors was smaller.

Bubbleu-Ambiguity showed a significantly lower perceived error rate compared to the baseline across all four error types (e.g., 11.11% decrease from 90.28% to 79.17%). The most significant decrease appeared in the *Misclassification* error. In contrast, *Bubbleu-Transparency* resulted in a higher perceived error rate for *Missing detection*, *False detection*, and *Misclassification* errors. The results for *Bubbleu-Controllability* differed across error types. The report rate of *Mislocalization* errors was lower than the other error types (except for *Bubbleu-Ambiguity* and *Bubbleu-Controllability* where error handling design was included), while *Missing detection* and *False detection* errors were frequently reported.

6.2.3 Gameplay experience and perceived task load. Finally, we investigated the perceived workload of the game by analyzing the perceived task load survey results collected from the free gameplay session (Figure 6). The reported result did not significantly differ between the baseline and the improved design. Two factors of *Bubbleu-Ambiguity* showed statistical difference: the self-reported

Table 6: Effectiveness of three design variants during free gameplay. Total number of attempts of interaction, successful attempts, failed attempts, total errors, and occurrences of different types of errors (* indicates p < 0.05.)

Version	# Intended Attempts	# Success	Success Rate	# Total Errors	# Missing Detection	# Mis-localization	# Mis-classification	# False Detection
Ambiguity	15.11	*11.78	*77.96	4.56	2.33	1	0.67	0.56
Baseline	13.56	*7.89	*58.19	6.56	3.78	1.11	0.78	0.89
Transparency	14.5	*11.7	80.69	2.9	1.8	0.8	0.2	0.1
Baseline	13.3	*9.6	72.18	4.2	2.1	1	0.6	0.5
Controllability	15.27	*11	72.04	4.91	2.27	1.09	0.91	0.64
Baseline	13.91	*9.55	68.66	5.36	2.18	1.09	1.09	1

Table 7: Perceived error rates in controlled gameplay.

Played Version	Missing Detection	False Detection	Misclassification	Mislocalization
Ambiguity	79.17	87.5	20.83	41.67
Baseline	91.67	91.67	50	37.5
Transparency	100	100	87.5	62.5
Baseline	100	94.44	79.17	70.83
Controllability	100	87.5	41.67	54.17
Baseline	95.83	91.67	91.67	37.5

task load related to the *Temporal Demand* (which indicates the amount of time pressure felt by the individual due to the pace of the task) and the *Performance* (which indicates the individual perception at how well they had achieved the goals of the task). This implies that the players had finished tasks with better performance when playing *Bubbleu-Ambiguity* but with higher temporal demand, which may serve as one hidden reason why the players did not prefer *Bubbleu-Ambiguity*.

6.2.4 User preference on designs. We now report the semi-structured interview results regarding the users' preference toward each design.

Bubbleu-Ambiguity. The participants showed mixed opinions about their preference for this design. Several participants (P3-P5, P9) mentioned that the ambiguous interaction of *observe* frustrated their intention to make an interaction: *"I felt confused when I expected an interaction, but the pet ended up with only observing."* (P3), *"It didn't seem sufficient when the pet's action finished after observation. I waited to see if there would be some other interaction but nothing happened."* (P9). Several other participants (P1, P11) mentioned that they disliked ambiguous expressions of object type: *"I was doubtful whether I was correct when the pet did not recognize the specific type of object."* (P1), *"I felt more fun with the interaction distributing diverse object types."* (P11). Participants who preferred *Bubbleu-Ambiguity* (P7, P8, P10, P12) answered that they preferred accuracy to the diversity of the interaction. Two of them mentioned the error rate while telling the preference: *"The first version (Bubbleu-Ambiguity) was better. It was better to show that the pet noticed something rather than doing some wrong interaction."* (P7) and *"I thought the second version(Bubbleu-Baseline) had many errors.*

I want the pet to rather observe instead of interact with a wrong object." (P8). The recorded gameplay video of the free study reveals that both players had experienced less rate of error in *Bubbleu-Ambiguity* (27.78%, 44.44% each) than *Bubbleu-Baseline* (54.17%, 50% each). Note that several participants used the direct term "observe" because the pet's dialogue during the observe interaction mentions that it is "observing" the object.

Bubbleu-Transparency. Overall, participants preferred the *Bubbleu-Transparency* design. Participants liked the additional information on the detection results (P15-19, P22-24): *"Since some interactions could be done with errors, it was useful to understand the object detection is correctly done or not."* (P19). They also liked the pet's advice on how to make detection better (P14-19, P22-24): *"I could correct the detection when it told me to move the camera around."* (P15). However, several participants (P13-14, P20-21) indicated that they did not feel the additional information was helpful: *"I am not sure if the confidence score information is necessary. The object was sometimes detected properly even when the confidence was low"* (P13), *"Whether or not I could check the result, I still cannot improve the accuracy of the system. I don't think the information is needed"* (P21).

Bubbleu-Controllability. The preference for *Bubbleu-Controllability* was interesting as every participant who played *Bubbleu-Controllability* liked the feature of threshold control. 6 (P25, P27-28, P30-32) mentioned that they preferred wrong interactions happening more than failing to interact: *"Failing in interaction in the first gameplay (Bubbleu-Baseline) frustrated me a lot. The second one (Bubbleu-Controllability) felt better when I had a chance to make the object detection more sensitive when the target object was not detected as I intended."* (P27). Some participants (P26, P34) explained the reason in relation to the correction feedback feature: *"I prefer to get wrong*

Table 8: User preference for three design variants vs. *Bubbleu-Baseline*.

Played Version	Design Feature	Counts	Quotes
Ambiguity	Different levels of interaction, *observe* at low confidence	4/12	*I expected some interaction but the pet ended up only observing.*
	Observe randomly when no detection happens	2/12	*I was confused when the pet continued observing even when I did not give any object.*
Transparency	Detection report	8/12	*I felt successful when the object I intended was detected with a high score.*
	Guide to recalibrate when no detection happens	9/12	*I did not notice the design at the time but I think showing the guidance is far better than none.*
Controllability	Ask for correction at low confidence	7/12	*It would be nice if the accuracy becomes higher in the future, but answering the validation question during the gameplay was cumbersome for me.*
	Ability to control threshold	12/12	*It is needed because it is very annoying when I consistently fail in detection.*

interaction rather than to fail the detection. Besides, I can tell the pet about the right interaction, so it is okay to be wrong." (P26).

7 out of 12 participants (P25-28, P30, P32, P34) preferred the other sub-design (the request for correction upon low confidence). Some participants (P25-26, P34) liked the ability to give feedback for the wrong detection results: *"The pet asked me to teach the correct interaction when it was confused between the object."* (P25) Some of them also mentioned that asking for correction made the pet feel more natural (P25, P27): *"I felt more like breeding the young pet when I taught it."* (P25). 6 participants (P26, P28, P30, P32, P34-35) mentioned the expectation of better accuracy: *"I liked giving the feedback to the pet in that my feedback would increase the accuracy in the future."* (P30). In contrast, 4 participants (P29, P31, P33, P36) mentioned that they disliked the feature in that the flow of the gameplay is disturbed: *"I think the feature is not needed. It only disturbs me from continuing the interaction. I would rather like to verify the result before the interaction starts."* (P29).

When asked to choose the preferred interaction design between the baseline and the improved design candidate, some participants mentioned the errors that frustrated the gameplay to explain the reason for the preference. For example, one participant preferred *Bubbleu-Ambiguity* and stated that *"Ending up by only observing is still better to misunderstanding the object."* (P8), while another participant reported that they preferred *Bubbleu-Baseline* mentioning that *"I was confused when the pet continued observing when I did not give any object."* (P6). Similarly, several participants who played *Bubbleu-Controllability* (P27, P30) mentioned they liked the design to control the threshold level in relation to the error types: *"Even if it gets wrong, I'd like to have threshold control. Wrong interaction felt better than nothing happens in my experience."* (P30).

Besides the preference for individual interaction designs, many participants (P1, P7-8, P9, P11, P13, P21, P23, P35-36) stated that they preferred the overall experience of one version over the other by explicitly referring to the perceived error rate of the gameplay. For example, *"I preferred the first version because it had less error than the second one"* (P8), or *"I liked the second version. The object was detected faster than the prior one."* (P21).

6.2.5   Other observations. **Relationship between player behaviors and error rates.** There was a bidirectional relationship between the error rates and player behaviors. During the free gameplay study, participants' way of manipulating the objects and the device affected the rate of errors. Commonly, the object is detected successfully when the object was positioned in the center of the screen and not occluded or cropped outside the scene [85]. Participants who tended to manipulate the objects closer to the device or who moved the camera continuously around the object could detect the object in a short time. In contrast, participants who tended to position the object far from the ideal scene (e.g., placing the object far away from the camera or occluding the object with their hand) experienced more errors.

The fluctuating error rates between participants also incurred different player behaviors, which made an impact on the overall error rate again. We observed that several participants developed their own ways to get object detection to become successful. For example, one participant (P35) figured out that an apple was easier to detect accurately and then attempted to interact only with apples. One participant (P8) failed to interact with the donut several times until they succeeded in the interaction when they rotated the donut upside down by accident. Afterward, they started rotating the donut upside down consistently. Another participant (P26) consistently put the objects right in front of the camera after they realized that this increased the success rate.

The strategy to recover from the failure also resulted in different error rates. For example, one participant (P2) immediately changed the object when they failed to detect the object, while another participant (P4) kept attempting to detect the same object multiple times until they finally succeeded. The rate of error differs largely between the two participants, as the latter participant experienced continuous *Missing detection* error before the success, while the prior participant did the interaction successfully on the second attempt.

Players' interpretation of uncertainty. When reporting the type of error that occurred during the user study, participants often misinterpreted the actual type of error as some other error. The most frequent confusion was understanding a *False detection* error as a *Misclassification* or a *Mislocalization* error. 17 participants (P1-2,
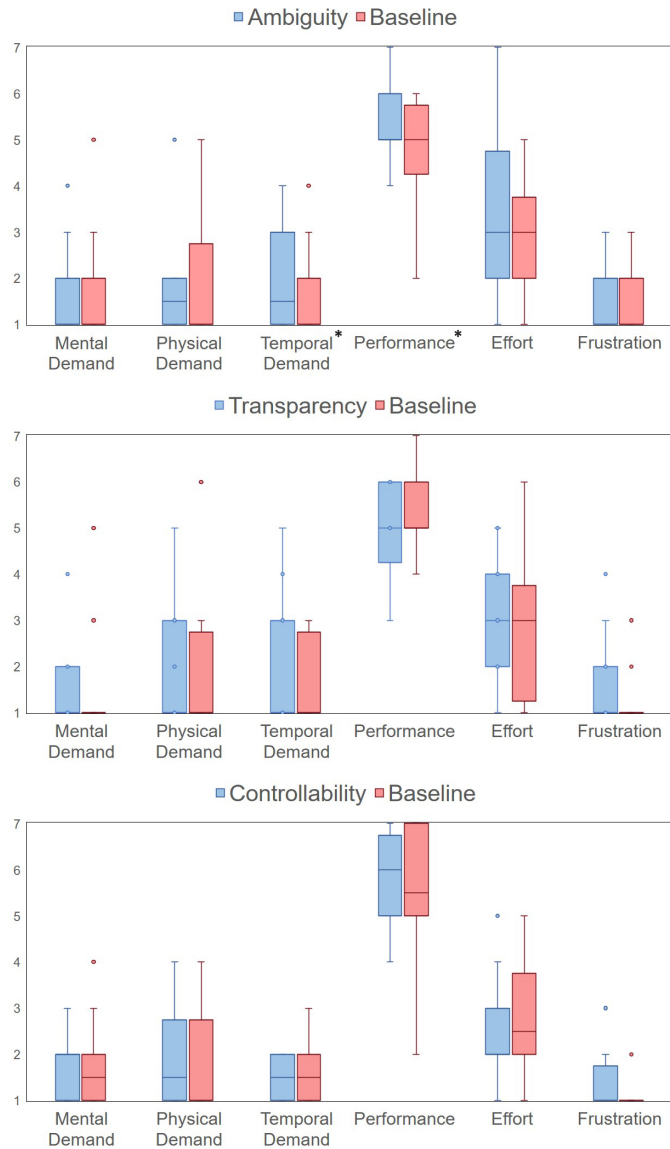
Figure 6: Perceived workload using 7-point Likert scales (* indicates p < 0.05). For Performance, a higher value implies a better experience. For the other components, the lower the better.

P6, P8-12, P17-18, P20, P22-24, P28-29, P36) mentioned that they had witnessed a *Misclassification* and *Mislocalization* error such as *"I tried to give a bottle to wash, but the pet tried to eat it at the other position"* (P11) or *"It should not have recognized my hand as a banana"* (P20). When a *False detection* error occurs at the same time when the player is manipulating an object, the pet would start unintended interaction at a different position (i.e., it interacts with a location where no object actually exists). The participants often misinterpreted this situation as both the position and the type of object were detected wrong.

Some of the participants (P1, P4, P16, P25, P27, P30) tended to understand the behavior of the system in relation to the narrative components of the pet. Some of them (P1, P4, P25, P27) mentioned the consistency of the detection result in relation to the preference of the pet: *"It seems that the pet wants to eat an apple. It is ignoring the donut."* (P4). One participant mentioned the background story of the pet (which is shown at the beginning of the game) when explaining their experience. They said that it is natural that the pet confuses the objects because it is *"a baby rabbit from the other planet"*: *"I was okay when it confused the objects because it's so young to distinguish the objects. What I felt wrong was when it totally missed an object."* (P16).

Players' expectation towards the AI-based system. Several participants (P2, P26, P28-29, P35) mentioned that they expected the pet to learn from experience: *"Does it learn the object when it observes? I think it is unsure what the object is."* (P2 while playing *Bubbleu-Ambiguity*). Some participants mentioned that the accuracy of the detection increased over time due to learning: *"As I taught the pet about the wrong interaction, the accuracy of the detection was better in the later part of the game."* (P26, after playing *Bubbleu-Controllability*), even though the object detection accuracy was not changed while they were playing.

7 DISCUSSION

In this section, we further discuss the results of our studies on three design variants. Also, based on the findings of our overall design study, we present general design implications to improve errors in object detection-based game interaction.

7.1 Impact of the Three Design Candidates

7.1.1 Ambiguity synergies with game aesthetics and reduces perceived errors. The results showed that the *Bubbleu-Ambiguity* variant reduced perceived error rates in both the free (Table 6) and controlled gameplay sessions (Table 7). The reduced perceived error rates imply that *Bubbleu-Ambiguity* successfully hides minor errors within the ambiguous expression. As an extension of the prior works [14, 28, 75], we found that our design has the ability to reduce the negative experience of the uncertain system. In addition, designers could adjust narratives of the games [41, 87] to the desired expressions. In some cases, the ambiguous expression improves the aesthetic components of the games [17, 28]. We plan to revisit the potential of using ambiguity to improve errors in other types of games in future work.

However, our post-study interview (Table 8) revealed that many participants were confused when ambiguous interactions appeared instead of their intended interactions. This is consistent with prior studies that showed that ambiguous interactions often confuse the users by displaying results that could be interpreted in multiple ways [28]. We attempted to mitigate this confusion by providing the game rewards (e.g., ambiguous interaction due to low confidence rewarded the same experience points as high confidence results). However, this itself could not resolve the confusion.

Another interesting finding was that the players using *Bubbleu-Ambiguity* finished their tasks with better performance but with higher temporal demand (Figure 6) – we plan to investigate this in more detail in the future. We thus conclude that even though our

Bubbleu-Ambiguity variant could reduce the perceived error and increase the players' perceived game performance, we should adopt it carefully, as ambiguous expressions may confuse and annoy the players.

7.1.2 Transparency helps players to understand and avoid errors. The transparent expressions helped users to understand and overcome errors but also led to a higher rate of perceived errors. Specifically, *Bubbleu-Transparency* showed lower error rates during the free gameplay but resulted in the highest perceived error rate in the controlled gameplay. As *Bubbleu-Transparency* reveals the label of the detected object and the confidence level, the players using *Bubbleu-Transparency* could more accurately discover errors compared to the baseline. Interestingly, this increased the overall success rate in the free gameplay sessions, as the players could self-learn how to control the device and objects to obtain more accurate results. This self-efficacy related to the fixation of errors was also reflected in the preference for the *Bubbleu-Transparency* design. During the interview, several players mentioned preferring the additional information as a useful means to overcome errors.

Generally, in conventional games, errors are mostly intolerable by users and considered as something to avoid as much as possible [54, 81]. However, in line with recent studies on using transparency in AI-based systems [27, 58], we observed that making errors more perceivable also had a positive impact on the player experience. This can be interpreted that users prefer having a better understanding of the system when uncertainty is inevitable.

7.1.3 Players prefer to have controllability over the uncertain system. Our *Bubbleu-Controllability* design, which provides control of the detection confidence threshold, was preferred by all users (highest number out of the three designs, 12/12) in our study. The results were well aligned with the findings from prior works; controllability enhances usability and induces a positive user experience by allowing users to manipulate the system [6, 31].

In *Bubbleu-Controllability*, lowering the detection threshold inevitably led to making more false positives. Though, many users preferred at least some interaction, even if they were wrong, over no interactions at all. On the other hand, a few users from our study also mentioned that they were confused when wrong interactions happened sometimes. This was consistent with studies of other AI-based applications such as a scheduling assistant [47]. Thus, we posit that providing the ability to control stands as a trade-off between the number of interactions and the detection accuracy, but can satisfy user preferences.

Although the primary goal of *Bubbleu-Controllability* was to enhance the player experience that it was not targeting to decrease the error rate itself, *Bubbleu-Controllability* also resulted in lower perceived error rates during controlled gameplay sessions. We observed that participants reported errors correctly when the pet asked the verification questions due to low detection confidence. However, when the pet did not ask the question with relatively high confidence scores, they did not report the errors. These differences mainly arise due to the limitations of the confidence-based error estimation. Confidence values are good proxies for the accuracy of predictions, but not a perfect measure to cover various usage scenarios. Exploring alternatives to confidence values remains an important future research topic.

7.2 Design Implications

We now discuss various design implications for AR games that utilize object interaction as an interaction mechanism.

7.2.1 Avoid stopping players from interacting. Even though it is ideal to build games without any errors, a certain percentage of errors is inevitable, especially for AI-enabled games. In such conditions, we noticed that the games should be still playable with errors. Filtering of incorrect results needs to be done conservatively. For example, the player's preference for the threshold control of *Bubbleu-Controllability* suggests that participants prefer interactions happening even when the subsequent interactions are not quite precise. Participants who played *Bubbleu-Ambiguity* and *Bubbleu-Transparency* also preferred continuing their interactions rather than not being able to interact when the detection confidence was low. We thus recommend that any confidence thresholds should be carefully configured with a primary focus on reducing or eliminating the players' inability to interact.

7.2.2 Good narrative can be used to excuse uncertainty. In Section 6.2.5, we observed that the narrative of the game serves as an explanation for the uncertainty and allows players to accept the failures. The narrative is one of the unique features of games [41, 71, 87]. In addition to aesthetic pleasure, game narratives also provide reasons for the various in-game behaviors [87]. Narratives have the potential to gracefully explain the uncertainty of the system. In this study, we did not include narratives as a main design candidate concept due to the difficulty of controlling narratives in our experiments. For future research, we suggest that adequate narratives could be included to compensate for in-game errors.

7.2.3 Provide time for players to verify in-game interactions. We suggest designing slower game interaction modalities when developing games with an object detection-based mechanic to ensure the AI inference system has sufficient time to interpret real-world scenes. Bubbleu does this by preserving additional time to detect the scene over multiple frames during the animation sequences. A large portion of *False detection* and *Missing detection* errors mainly occur in short frames between the correct frames. The system can easily detect and eliminate errors in inconsistent frames in a slow and steady scene. In contrast, games where the scene changes quickly (e.g., a game that is played while running) would have insufficient time to confirm the detection. Moreover, in situations where the device itself moves quickly, motion blur in the images would worsen the detection quality.

7.3 Impact of Object Types on Detection Errors

To quantitatively identify the occurrence of different object detection error types when using different objects, we collected 1500 seconds of video of the research team playing Bubbleu. We ran object detection on each frame of the video and investigated errors in every detected bounding box. Table 9 shows the breakdown of object detection error rates by type of object. The results show that the four different types of errors occur with different frequencies and at different rates depending on the type of object being detected. The cup object had the lowest error rate, having fewer *Missing detection* errors (10.39 %, compared to 22.62 % on average) but more *Mislocalization* errors. In contrast, donuts had much higher error

Table 9: Error rates in detecting different objects

Object type	Total frames	Correct	Miss	False	Loc	Cls
Apple	1276	62.62	17.79	4.47	1.25	13.87
Banana	1206	64.93	18.41	4.06	0.33	12.27
Orange	1161	60.29	18.09	7.24	0.95	13.44
Donut	1242	31.08	49.52	4.99	2.25	12.16
Cup	1222	71.69	10.39	4.5	4.66	8.76
Bottle	1046	64.44	20.75	2.29	1.24	11.28
Total	7153	58.97	22.62	4.63	1.8	11.98

rates than other object types. This implies that the choice of objects used can have a high impact on the end-user experience.

7.4 Limitations and Future Work

Scaling the user study. In our study, the number, and diversity of participants per design were relatively small (36 participants divided into three groups). Also, the participants were recruited from the same country and selected from a small pool (using an online survey) over a short period of 3 weeks. In our future work, we plan to investigate the player experience with more diverse participant demographics and with a longer duration study to understand the learning effects.

Diversifying the designs. In this work, we investigated just three distinct design concepts from a much larger design space. Even for the three design candidates, many additional variants and implementation options are available. For example, in *Bubbleu-Transparency*, we can provide object detection status and confidence level information in different ways (e.g., by providing saliency maps showing how the camera captured the scene and which pixels contributed the most to the detection outcome [45]). Furthermore, we can allow more fine-grained control in *Bubbleu-Controllability* by allowing the user to set different detection confidence thresholds per object type. We could also extend our system to other design concepts (some are listed in Table 10 in the Appendix). Lastly, studying the effects of concurrently applying multiple design concepts remains an important future research direction.

Extending to other games and AI models. We also plan to study the implications of our error-handling strategies for other types of games and AI-based interactions. In this work, we focused on a single game (i.e., a pet breeding game) with object detection as an underlying game mechanic. It will be interesting to explore a similar set of research questions for other types of games (e.g., fast-pacing shooting games), interaction categories (e.g., multi-object interaction), and other underlying AI models (e.g., segmentation, 3D object detection, or generation).

8 CONCLUSION

We presented the design of Bubbleu to investigate the potential of object detection as an AR game interaction method – in particular, how game design strategies affect the user perception of underlying object detection errors. Through an empirical design improvement process coupled with a user study, we showed the potential of three different game design concepts - *Ambiguity*, *Transparency*, and

Controllability to achieve better player experience with uncertain AI-based interactions. We observed that *Ambiguity* and *Transparency* improve the success rate of interaction, but in completely different ways; *Ambiguity* hides the errors perceived by the user while *Transparency* explicitly exposes the errors and helps users learn how to interact appropriately. We also found that the players preferred *Transparency* and *Controllability* over *Ambiguity*, indicating that game experiences are less affected by errors when given higher controllability and diversity of interactions. We believe that our findings will serve as a useful design material for the design of future AR games that use object detection as a key mechanism.

REFERENCES

[1] [n. d.]. AI ethics in action. https://www.ibm.com/downloads/cas/4DPJK92W. Accessed: 2022-12-14.
[2] [n. d.]. Designing the UI and user experience of a machine learning app. https://developer.apple.com/design/human-interface-guidelines/technologies/machine-learning/introduction. Accessed: 2022-12-14.
[3] [n. d.]. People+AI Guidebook. https://pair.withgoogle.com/guidebook. Accessed: 2022-12-14.
[4] Abdullah Abuolaim, Abhijith Punnappurath, and Michael S. Brown. 2018. Revisiting Autofocus for Smartphone Cameras. In *Proceedings of the European Conference on Computer Vision (ECCV)*.
[5] Amina Adadi and Mohammed Berrada. 2018. Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access* 6 (2018), 52138–52160. https://doi.org/10.1109/ACCESS.2018.2870052
[6] Amina Adadi and Mohammed Berrada. 2018. Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). *IEEE access* 6 (2018), 52138–52160.
[7] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N Bennett, Kori Inkpen, et al. 2019. Guidelines for human-AI interaction. In *Proceedings of the 2019 chi conference on human factors in computing systems*. 1–13.
[8] Mike Ananny and Kate Crawford. 2018. Seeing without knowing: Limitations of the transparency ideal and its application to algorithmic accountability. *new media & society* 20, 3 (2018), 973–989.
[9] Gagan Bansal, Besmira Nushi, Ece Kamar, Daniel S Weld, Walter S Lasecki, and Eric Horvitz. 2019. Updates in human-ai teams: Understanding and addressing the performance/compatibility tradeoff. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 2429–2437.
[10] Evan Barba, Blair MacIntyre, and Elizabeth D Mynatt. 2012. Here we are! Where are we? Locating mixed reality in the age of the smartphone. *Proc. IEEE* 100, 4 (2012), 929–936.
[11] Tom Beigbeder, Rory Coughlan, Corey Lusher, John Plunkett, Emmanuel Agu, and Mark Claypool. 2004. The effects of loss and latency on user performance in unreal tournament 2003®. In *Proceedings of 3rd ACM SIGCOMM workshop on Network and system support for games*. 144–151.
[12] Mark Billinghurst, Adrian Clark, and Gun Lee. 2015. A survey of augmented reality. (2015).
[13] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. 2020. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934* (2020).
[14] Kirsten Boehner and Jeffrey T Hancock. 2006. Advancing ambiguity. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*. 103–106.
[15] Daniel Bolya, Sean Foley, James Hays, and Judy Hoffman. 2020. Tide: A general toolbox for identifying object detection errors. (2020), 558–573.

[16] Yining Cao, Hariharan Subramonyam, and Eytan Adar. 2022. VideoSticker: A Tool for Active Viewing and Visual Note-Taking from Videos. In *27th International Conference on Intelligent User Interfaces* (Helsinki, Finland) *(IUI '22)*. Association for Computing Machinery, New York, NY, USA, 672–690. https://doi.org/10.1145/3490099.3511132

[17] Diane Carr, Martin Oliver, and Andrew Burn. 2010. Learning, teaching and ambiguity in virtual worlds. In *Researching learning in virtual worlds*. Springer, 17–30.

[18] Devleena Das, Siddhartha Banerjee, and Sonia Chernova. 2021. Explainable AI for Robot Failures: Generating Explanations That Improve User Assistance in Fault Recovery. In *Proceedings of the 2021 ACM/IEEE International Conference on Human-Robot Interaction* (Boulder, CO, USA) *(HRI '21)*. Association for Computing Machinery, New York, NY, USA, 351–360. https://doi.org/10.1145/3434073.3444657

[19] Martin Dechant, Ian Stavness, Aristides Mairena, and Regan L Mandryk. 2018. Empirical evaluation of hybrid gaze-controller selection techniques in a gaming context. In *Proceedings of the 2018 Annual Symposium on Computer-Human Interaction in Play*. 73–85.

[20] Dazhen Deng, Jiang Wu, Jiachen Wang, Yihong Wu, Xiao Xie, Zheng Zhou, Hui Zhang, Xiaolong (Luke) Zhang, and Yingcai Wu. 2021. EventAnchor: Reducing Human Interactions in Event Annotation of Racket Sports Videos. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 73, 13 pages. https://doi.org/10.1145/3411764.3445431

[21] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 248–255. https://doi.org/10.1109/CVPR.2009.5206848

[22] Ansgar E Depping and Regan L Mandryk. 2017. Why is this happening to me? how player attribution can broaden our understanding of player experience. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 1040–1052.

[23] Sebastian Deterding, Dan Dixon, Rilla Khaled, and Lennart Nacke. 2011. From Game Design Elements to Gamefulness: Defining "Gamification". In *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments* (Tampere, Finland) *(MindTrek '11)*. Association for Computing Machinery, New York, NY, USA, 9–15. https://doi.org/10.1145/2181037.2181040

[24] Nicholas Diakopoulos. 2016. Accountability in algorithmic decision making. *Commun. ACM* 59, 2 (2016), 56–62.

[25] Fiona Draxler, Audrey Labrie, Albrecht Schmidt, and Lewis L Chuang. 2020. Augmented reality to enable users in learning case grammar from their real-world interactions. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–12.

[26] Carlo Fabricatore. 2007. Gameplay and game mechanics: a key to quality in videogames. (2007).

[27] Heike Felzmann, Eduard Fosch Villaronga, Christoph Lutz, and Aurelia Tamò-Larrieux. 2019. Transparency you can trust: Transparency requirements for artificial intelligence between legal norms and contextual concerns. *Big Data & Society* 6, 1 (2019), 2053951719860542.

[28] William W Gaver, Jacob Beaver, and Steve Benford. 2003. Ambiguity as a resource for design. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 233–240.

[29] Katy Ilonka Gero, Zahra Ashktorab, Casey Dugan, Qian Pan, James Johnson, Werner Geyer, Maria Ruiz, Sarah Miller, David R Millen, Murray Campbell, et al. 2020. Mental Models of AI Agents in a Cooperative Game Setting. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–12.

[30] Tarleton Gillespie. 2014. The relevance of algorithms. *Media technologies: Essays on communication, materiality, and society* 167, 2014 (2014), 167.

[31] Miriam Greis, Hendrik Schuff, Marius Kleiner, Niels Henze, and Albrecht Schmidt. 2017. Input controls for entering uncertain data: Probability distribution sliders. *Proceedings of the ACM on Human-Computer Interaction* 1, EICS (2017), 1–17.

[32] David Gunning. 2017. Explainable artificial intelligence (xai). *Defense advanced research projects agency (DARPA), nd Web* 2, 2 (2017), 1.

[33] Neilly H. Tan, Richmond Y. Wong, Audrey Desjardins, Sean A. Munson, and James Pierce. 2022. Monitoring Pets, Deterring Intruders, and Casually Spying on Neighbors: Everyday Uses of Smart Home Cameras. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 617, 25 pages. https://doi.org/10.1145/3491102.3517617

[34] Sandra G Hart and Lowell E Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Advances in psychology*. Vol. 52. Elsevier, 139–183.

[35] Edward Helmore. [n. d.]. Tesla's self-driving technology fails to detect children in the road, group claims. *The Guardian* ([n. d.]). https://www.theguardian.com/technology/2022/aug/09/tesla-self-driving-technology-safety-children

[36] Tristan Henderson. 2001. Latency and user behaviour on a multiplayer game server. In *International Workshop on Networked Group Communication*. Springer, 1–13.

[37] Oliver Hohlfeld, Hannes Fiedler, Enric Pujol, and Dennis Guse. 2016. Insensitivity to network delay: minecraft gaming experience of casual gamers. In *2016 28th International Teletraffic Congress (ITC 28)*, Vol. 3. IEEE, 31–33.

[38] Derek Hoiem, Yodsawalai Chodpathumwan, and Qieyun Dai. 2012. Diagnosing error in object detectors. (2012), 340–353.

[39] Robin Hunicke, Marc LeBlanc, and Robert Zubek. 2004. MDA: A formal approach to game design and game research. 4, 1 (2004), 1722.

[40] Yasha Iravantchi, Mayank Goel, and Chris Harrison. 2020. Digital Ventriloquism: Giving Voice to Everyday Objects. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–10. https://doi.org/10.1145/3313831.3376503

[41] Henry Jenkins. 2004. Game design as narrative architecture. *Computer* 44, 3 (2004), 118–130.

[42] Seokbin Kang, Ekta Shokeen, Virginia L Byrne, Leyla Norooz, Elizabeth Bonsignore, Caro Williams-Pierce, and Jon E Froehlich. 2020. ARMath: augmenting everyday life with math learning. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–15.

[43] Anjali Khurana, Parsa Alamzadeh, and Parmit K Chilana. 2021. ChatrEx: Designing explainable chatbot interfaces for enhancing usefulness, transparency, and trust. In *2021 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, 1–11.

[44] Antino Kim, Mochen Yang, and Jingjing Zhang. 2022. When Algorithms Err: Differential Impact of Early vs. Late Errors on Users' Reliance on Algorithms. *ACM Trans. Comput.-Hum. Interact.* (aug 2022). https://doi.org/10.1145/3557889 Just Accepted.

[45] Jacob Kittley-Davies, Ahmed Alqaraawi, Rayoung Yang, Enrico Costanza, Alex Rogers, and Sebastian Stein. 2019. Evaluating the Effect of Feedback from Different Computer Vision Processing Stages: A Comparative Lab Study. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300273

[46] René F Kizilcec. 2016. How much information? Effects of transparency on trust in an algorithmic interface. In *Proceedings of the 2016 CHI conference on human factors in computing systems*. 2390–2395.

[47] Rafal Kocielnik, Saleema Amershi, and Paul N. Bennett. 2019. Will You Accept an Imperfect AI? Exploring Designs for Adjusting End-User Expectations of AI Systems. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3290605.3300641

[48] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger (Eds.), Vol. 25. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[49] Kuno Kurzhals, Fabian Göbel, Katrin Angerbauer, Michael Sedlmair, and Martin Raubal. 2020. A View on the Viewer: Gaze-Adaptive Captions for Videos. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3313831.3376266

[50] Taeahn Kwon, Minkyung Jeong, Eon-Suk Ko, and Youngki Lee. 2022. Captivate! Contextual Language Guidance for Parent–Child Interaction. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 219, 17 pages. https://doi.org/10.1145/3491102.3501865

[51] Samuli Laato, Miika Tiainen, AKM Najmul Islam, and Matti Mäntymäki. 2022. How to explain AI systems to end users: a systematic literature review and research agenda. *Internet Research* 32, 7 (2022), 1–31.

[52] Bruno Lepri, Nuria Oliver, Emmanuel Letouzé, Alex Pentland, and Patrick Vinck. 2018. Fair, transparent, and accountable algorithmic decision-making processes. *Philosophy & Technology* 31, 4 (2018), 611–627.

[53] Ariel Levy, Monica Agrawal, Arvind Satyanarayan, and David Sontag. 2021. Assessing the Impact of Automated Suggestions on Decision Making: Domain Experts Mediate Model Errors but Take Less Initiative. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 72, 13 pages. https://doi.org/10.1145/3411764.3445522

[54] Chris Lewis, Jim Whitehead, and Noah Wardrip-Fruin. 2010. What went wrong: a taxonomy of video game bugs. In *Proceedings of the fifth international conference on the foundations of digital games*. 108–115.

[55] Wei Liang, Xinzhe Yu, Rawan Alghofaili, Yining Lang, and Lap-Fai Yu. 2021. Scene-Aware Behavior Synthesis for Virtual Pets in Mixed Reality. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 472, 12 pages. https://doi.org/10.1145/3411764.3445532

[56] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. (2014), 740–755.

[57] Martin Lindvall, Claes Lundström, and Jonas Löwgren. 2021. Rapid Assisted Visual Search: Supporting Digital Pathologists with Imperfect AI *(IUI '21)*. Association for Computing Machinery, New York, NY, USA, 504–513. https://doi.org/10.1145/3397481.3450681

[58] Zachary C Lipton. 2018. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue* 16, 3 (2018), 31–57.

[59] Shengmei Liu and Mark Claypool. 2022. The Impact of Latency on Navigation in a First-Person Perspective Game. In *CHI Conference on Human Factors in Computing Systems*. 1–11.

[60] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. 2016. Ssd: Single shot multibox detector. (2016), 21–37.

[61] Michael Long and Carl Gutwin. 2019. Effects of Local Latency on Game Pointing Devices and Game Pointing Tasks. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300438

[62] Luca Longo, Randy Goebel, Freddy Lecue, Peter Kieseberg, and Andreas Holzinger. 2020. Explainable artificial intelligence: Concepts, applications, research challenges and visions. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*. Springer, 1–16.

[63] Matt McFarland. [n. d.]. Tesla Autopilot's safety questioned after latest fatal motorcycle crash. *CNN Business* ([n. d.]). https://edition.cnn.com/2022/10/17/business/tesla-motorcycle-crashes-autopilot/index.html

[64] Sina Mohseni, Niloofar Zarei, and Eric D Ragan. 2021. A multidisciplinary survey and framework for design and evaluation of explainable AI systems. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 11, 3-4 (2021), 1–45.

[65] Sudipto Kumar Mondal, Indraneel Mukhopadhyay, and Supreme Dutta. 2020. Review and Comparison of Face Detection Techniques. In *Proceedings of International Ethical Hacking Conference 2019*, Mohuya Chakraborty, Satyajit Chakrabarti, and Valentina E. Balas (Eds.). Springer Singapore, Singapore, 3–14.

[66] Chelsea Myers, Anushay Furqan, Jessica Nebolsky, Karina Caro, and Jichen Zhu. 2018. Patterns for how users overcome obstacles in voice user interfaces. In *Proceedings of the 2018 CHI conference on human factors in computing systems*. 1–7.

[67] Mahsan Nourani, Joanie King, and Eric Ragan. 2020. The Role of Domain Expertise in User Trust and the Impact of First Impressions with Intelligent Systems. *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing* 8, 1 (Oct. 2020), 112–121. https://doi.org/10.1609/hcomp.v8i1.7469

[68] Mahsan Nourani, Chiradeep Roy, Jeremy E Block, Donald R Honeycutt, Tahrima Rahman, Eric Ragan, and Vibhav Gogate. 2021. Anchoring Bias Affects Mental Model Formation and User Reliance in Explainable AI Systems. In *26th International Conference on Intelligent User Interfaces* (College Station, TX, USA) *(IUI '21)*. Association for Computing Machinery, New York, NY, USA, 340–350. https://doi.org/10.1145/3397481.3450639

[69] Ankit B Patel, Minh Tan Nguyen, and Richard Baraniuk. 2016. A Probabilistic Framework for Deep Learning. In *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (Eds.), Vol. 29. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2016/file/c70daf247944fe3add32218f914c75a6-Paper.pdf

[70] Felix Putze, Dennis Küster, Timo Urban, Alexander Zastrow, and Marvin Kampen. 2020. Attention Sensing through Multimodal User Modeling in an Augmented Reality Guessing Game. In *Proceedings of the 2020 International Conference on Multimodal Interaction*. 33–40.

[71] Hua Qin, Pei-Luen Patrick Rau, and Gavriel Salvendy. 2009. Measuring player immersion in the computer game narrative. *Intl. Journal of Human–Computer Interaction* 25, 2 (2009), 107–133.

[72] Emilee Rader, Kelley Cotter, and Janghee Cho. 2018. Explanations as mechanisms for supporting algorithmic transparency. In *Proceedings of the 2018 CHI conference on human factors in computing systems*. 1–13.

[73] Jacek Rumiński, Adam Bujnowski, Tomasz Kocejko, Jerzy Wtorek, Alexey Andrushevich, Martin Biallas, and Rolf Kistler. 2016. Performance Analysis of Interaction between Smart Glasses and Smart Objects Using Image-Based Object Identification. *International Journal of Distributed Sensor Networks* 12, 3 (2016), 6254827. https://doi.org/10.1155/2016/6254827 arXiv:https://doi.org/10.1155/2016/6254827

[74] Valentin Schwind, Sven Mayer, Alexandre Comeau-Vermeersch, Robin Schweigert, and Niels Henze. 2018. Up to the finger tip: The effect of avatars on mid-air pointing accuracy in virtual reality. In *Proceedings of the 2018 Annual Symposium on Computer-Human Interaction in Play*. 477–488.

[75] Phoebe Sengers and Bill Gaver. 2006. Staying open to interpretation: engaging multiple meanings in design and evaluation. In *Proceedings of the 6th conference on Designing Interactive systems*. 99–108.

[76] Miguel Sicart. 2008. Defining game mechanics. *Game studies* 8, 2 (2008), 1–14.

[77] Shubham Srivastava, Ajay Verma, and Shekhar Sharma. 2022. Optical Character Recognition Techniques: A Review. In *2022 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*. 1–6. https://doi.org/10.1109/SCEECS54111.2022.9740911

[78] Penelope Sweetser and Peta Wyeth. 2005. GameFlow: a model for evaluating player enjoyment in games. *Computers in Entertainment (CIE)* 3, 3 (2005), 3–3.

[79] Mingxing Tan, Ruoming Pang, and Quoc V Le. 2020. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10781–10790.

[80] Brad Templeton. [n. d.]. Tesla Removes Ultrasonic Sensors In Bold Move That Cripples Features But Promises To Restore Them. *Forbes* ([n. d.]). https://www.forbes.com/sites/bradtempleton/2022/10/17/tesla-removes-ultrasonic-sensors-in-bold-move-that-cripples-features-but-promises-to-restore-them/?sh=6bb35c6e4949

[81] Luke Thominet. 2018. Bugs and Emotion: A Content Analysis of Quality Assurance Player Feedback. In *Proceedings of the 36th ACM International Conference on the Design of Communication*. 1–2.

[82] Maggie Tilman. [n. d.]. Amazon Go and Amazon Fresh: How the 'Just walk out' tech works. *Pocket-lint* ([n. d.]). https://www.pocket-lint.com/gadgets/news/amazon/139650-what-is-amazon-go-where-is-it-and-how-does-it-work

[83] Mike Treanor, Alexander Zook, Mirjam P Eladhari, Julian Togelius, Gillian Smith, Michael Cook, Tommy Thompson, Brian Magerko, John Levine, and Adam Smith. 2015. AI-based game design patterns. (2015).

[84] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. 2022. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207.02696* (2022).

[85] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip H. S. Torr. 2018. Fast Online Object Tracking and Segmentation: A Unifying Approach. *CoRR* abs/1812.05050 (2018). arXiv:1812.05050 http://arxiv.org/abs/1812.05050

[86] Tianyi Wang, Xun Qian, Fengming He, Xiyun Hu, Ke Huo, Yuanzhi Cao, and Karthik Ramani. 2020. CAPturAR: An Augmented Reality Tool for Authoring Human-Involved Context-Aware Applications. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '20)*. Association for Computing Machinery, New York, NY, USA, 328–341. https://doi.org/10.1145/3379337.3415815

[87] Huaxin Wei. 2011. *Analyzing the game narrative: Structure and technique*. Ph. D. Dissertation. Communication, Art & Technology: School of Interactive Arts and Technology.

[88] Bernard Weiner. 2014. The attribution approach to emotion and motivation: History, hypotheses, home runs, headaches/heartaches. *Emotion Review* 6, 4 (2014), 353–361.

[89] Richard Wetzel, Lisa Blum, Wolfgang Broll, and Leif Oppermann. 2011. Designing mobile augmented reality games. In *Handbook of Augmented Reality*. Springer, 513–539.

[90] Richard Wetzel, Rod McCall, Anne-Kathrin Braun, and Wolfgang Broll. 2008. Guidelines for Designing Augmented Reality Games. In *Proceedings of the 2008 Conference on Future Play: Research, Play, Share* (Toronto, Ontario, Canada) *(Future Play '08)*. Association for Computing Machinery, New York, NY, USA, 173–180. https://doi.org/10.1145/1496984.1497013

[91] Qian Yang, Aaron Steinfeld, Carolyn Rosé, and John Zimmerman. 2020. Re-examining whether, why, and how human-AI interaction is uniquely difficult to design. In *Proceedings of the 2020 chi conference on human factors in computing systems*. 1–13.

[92] Nima Zargham, Johannes Pfau, Tobias Schnackenberg, and Rainer Malaka. 2022. "I Didn't Catch That, But I'll Try My Best": Anticipatory Error Handling in a Voice Controlled Game. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 153, 13 pages. https://doi.org/10.1145/3491102.3502115

[93] Wencan Zhang and Brian Y Lim. 2022. Towards Relatable Explainable AI with the Perceptual Process. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 181, 24 pages. https://doi.org/10.1145/3491102.3501826

[94] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. 2019. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems* 30, 11 (2019), 3212–3232.

[95] Jichen Zhu, Jennifer Villareale, Nithesh Javvaji, Sebastian Risi, Mathias Löwe, Rush Weigelt, and Casper Harteveld. 2021. Player-AI Interaction: What Neural Network Games Reveal About AI as Play. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 77, 17 pages. https://doi.org/10.1145/3411764.3445307

[96] Zhengxia Zou, Zhenwei Shi, Yuhong Guo, and Jieping Ye. 2019. Object detection in 20 years: A survey. *arXiv preprint arXiv:1905.05055* (2019).

A   DESIGN EXPLORATION WORKSHOP RESULTS

Table 10: Design ideas from the workshop

Related Concepts	Design
Narratives	The pet tells "I'm not hungry" when the confidence is low.
Narratives	The pet has bad sights that it cannot interact objects in a far distance.
Narratives	Express as if the pet prefers the object that are commonly detected in high accuracy(e.g., apple)
Narratives Randomness	Add pet behaviors to indicate that the pet is young and unskilled (e.g., falling down occasionally)
Ambiguity	Use ambiguous interaction expression instead of showing precise results (e.g., show 'fruit' icon instead of an 'apple' icon)
Ambiguity	Show detection results in different levels regarding the detection confidence (e.g., show 'eat' icon when confidence is low, show 'apple' when high)
Ambiguity	Add an interaction that can be done whether the error occurs
Transparency	Tell the players that detection is not being done when no object is detected for a long time period
Transparency	Show detection results before the interaction
Controllability	After the interaction, let players choose to retry in case it is wrong
Controllability	Let players to give feedbacks when the pet is wrong
Controllability	Enable players to force specific interactions with a button (e.g., force the pet to eat with an 'eat' button)
Guidance	Ask players to move objects around when the object is not detected
Emotion	Show additional message when a hand is detected (because the players feel large emotional fulfillment when touching the pet)
Challenge	Give better rewards when the detected confidence is high
Randomness	Distract the players when the error is predicted (e.g., another pet appears, the pet runs away ...)