“I Didn’t Catch That, But I’ll Try My Best”: Anticipatory Error Handling in a Voice Controlled Game

Conference Paper · April 2022

DOI: 10.1145/3491102.3502115

CITATION
1

READS
26

4 authors, including:

Nima Zargham
Universität Bremen
14 PUBLICATIONS **27** CITATIONS
SEE PROFILE

Johannes Pfau
University of California, Santa Cruz
25 PUBLICATIONS **162** CITATIONS
SEE PROFILE

Rainer Malaka
Universität Bremen
316 PUBLICATIONS **3,339** CITATIONS
SEE PROFILE

"I Didn't Catch That, But I'll Try My Best": Anticipatory Error Handling in a Voice Controlled Game

Nima Zargham
zargham@uni-bremen.de
Digital Media Lab
University of Bremen
Germany

Johannes Pfau
jpfau@uni-bremen.de
Digital Media Lab
University of Bremen
Germany

Tobias Schnackenberg
tschnack@uni-bremen.de
Digital Media Lab
University of Bremen
Germany

Rainer Malaka
malaka@uni-bremen.de
Digital Media Lab
University of Bremen
Germany

ABSTRACT

Advances in speech recognition, language processing and natural interaction have led to an increased industrial and academic interest. While the robustness and usability of such systems are steadily increasing, speech-based systems are still susceptible to recognition errors. This makes intelligent error handling of utmost importance for the success of those systems. In this work, we integrated anticipatory error handling for a voice-controlled video game where the game would perform a locally optimized action in respect to goal completion and obstacle avoidance, when a command is not recognized. We evaluated the user experience of our approach versus traditional, repetition-based error handling ($N = 34$). Our results indicate that implementing anticipatory error handling can improve the usability of a system, if it follows the intention of the user. Otherwise, it impairs the user experience, even when deciding for technically optimal decisions.

CCS CONCEPTS

• **Human-centered computing → Natural language interfaces**;
• **Applied computing → Computer games**.

KEYWORDS

Voice User Interfaces, Game Design, Error Handling, Speech-Based Systems, Voice-Controlled Game, Voice Interaction

ACM Reference Format:
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 *CHI Conference on Human Factors in Computing Systems (CHI '22), April 29-May 5, 2022, New Orleans, LA, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3491102.3502115

CHI '22, April 29-May 5, 2022, New Orleans, LA, USA

ACM ISBN 978-1-4503-9157-3/22/04…$15.00
https://doi.org/10.1145/3491102.3502115

1 INTRODUCTION

Voice user interfaces (VUIs) are gaining more and more attention in recent years due to the intuitive nature of their interaction. Speaking is a natural way of communication amongst humans and people find it easier to interact with technology that resembles some of their own characteristics [15]. Voice input is now a feature in many devices such as mobile phones, cars and home assistants. In their early days, VUIs were designed for handling few specialized tasks [55], but due to the advancements in the technology, they now can have a broad range of capabilities in performing various functions in different settings. Current VUIs are used for various purposes such as smart home control, scheduling, navigation, education, and entertainment. The technical aspects of the VUIs, as well as their usability and user experience (UX), have been the subject of extensive research in the recent years [21, 23, 25, 43, 44].

In order to integrate speech recognition, developers need to have a large repository of collected voice data so that the system has enough information to process different inflections and variations in different voices. If the product is aimed at the global market, different languages, accents, and dialects need to be considered to assure a better recognition system. On top of that, different forms of phrasing for a single command should be incorporated to allow for a more natural experience, underlining the issue that designing a satisfying experience with speech-based systems is a complex and difficult process.

Although this technology is steadily improving in various aspects, speech-based systems are still prone to recognition failures. Several elements such as hardware limitations, background noises and language barriers make designing voice interfaces a very complex and time-consuming task. Researchers believe that problems with speech recognition and limited functionality are the main reasons for disliking or not using voice systems [11]. Users have frequently reported that they find voice interaction disappointing or embarrassing, which lets such systems appear as unintelligent and immature [5, 11, 18, 42, 48]. This makes error handling a critical part of designing VUIs, which includes the situations where the system does not understand the user's command, the given command is out of context, or the command is misunderstood [39]. Several guidelines for designing fallback strategies have been proposed,

such as asking the user to repeat the command, redirecting the user to the tasks that the system can support, or presenting user options to correct their commands [12, 39, 51]. In some cases, the voice assistant (VA) falls back on humor in response to complex conversational input and commands that cannot be handled otherwise, which might be seen as sarcastic or entertaining [56].

Recently, this technology has gained considerable attention in the entertainment industry and video game companies have been adopting voice-activated services to their games. As speech recognition technology is improving rapidly and the number of available microphones in consumer gaming devices is growing every day, it leaves a great potential for using VUIs in games [2]. This allows voice-control to be used as an appealing and intuitive feature in video games to enhance the experience of the players. Speaking is a natural and enjoyable way of interacting, which can increase social presence within the game and make them more immersive [38, 76]. With the release of Microsoft Kinect in 2010, Xbox games in various genres such as *Mass Effect 3* [31], *FIFA 14* [30], *Forza Motorsport 5* [67], and *Ryse: Son of Rome* [27] took advantage of the voice interaction that was provided by Kinect. However, in most cases, voice input is an optional feature and not a core element of the game design.

Voice-activated games attempted to provide natural language input, but this experience has been frequently described as "uncomfortable" and "awkward" by players [29]. Video games are mainly goal-oriented activities, and players find enjoyment when they work towards this goal [35]. If the challenge is right, the players are in a state of flow [28]. The misrecognition of voice input in video games adds another layer of challenge on top of the game's existing obstacles, preventing players from reaching their goals and staying in the state of flow, which often results in player frustration. Moreover, studies have shown that once a recognition error occurs, the likelihood of having an error in the next intent increases [13, 60, 66]. One of the reasons for this is that, as more errors occur, user's patience runs out and frustration increases, which can lead to acoustic and language mismatches [13]. Previous research has shown that human operators often do not signal non-understandings, but rather try to advance the task by asking different questions, which generally led to a speedier recovery [63]. Similarly, for speech-based systems, researchers suggest that when non-understandings happen, instead of trying to repair the current problem, use an alternative dialog plan to advance the task [13].

On this basis, we designed a voice-controlled video game with the aim of investigating user experience with two different error handling methodologies. In this game, players control the game protagonist using voice commands. A between-subjects user study was conducted to compare traditional repetition-based error handling with a novel approach implementing anticipatory error handling within the game. In the control group, the game would notify the player of the recognition failure so that the player could repeat the command once again. With anticipatory error handling, if a command was not recognized, the game would proceed by performing a locally optimized action in respect to goal completion and obstacle avoidance without notifying the player about the recognition failure. In the scope of this work, when we refer to recognition failure, our interest lies in command recognition, which is a subset of natural language understanding (NLU). Nonetheless, the insights of this work might also hold for certain NLU issues.

In this study, we pursue the following research questions:

RQ1 Does performing a locally optimized game action in times of misrecognition lead to a measurably improved usability in a speech-based video game?

RQ2 What are the effects on player experience in terms of competency, autonomy, presence, and intuitive control, if error handling mechanisms decide for unintended actions?

Based on our design space and the existing literature, we developed the following hypotheses:

- *H1: Participants will observe a lower number of recognition errors in case of anticipatory error handling.*
- *H2: The anticipatory error handling will lead to a higher rating regarding:*
  (a) *players' perceived competence.*
  (b) *players' perceived autonomy.*
  (c) *players' perceived presence.*
  (d) *intuitiveness of the game controls.*

Our results showed significantly higher usability ratings for the anticipatory error handling, as well as a significantly lower number of perceived errors for this condition. Furthermore, this study contributes useful insights and implications on the user experience with recognition error handling in speech-based systems, most importantly the users' aversion to error handling that opposed their intention – even in cases of goal-directed and anticipated solutions.

2 RELATED WORK

Since the early success of voice and gesture in an interface with the "Put that There" system [14], voice user interfaces have been largely investigated by researchers in the field of HCI. In this section, we provide a summary of the previous literature on speech-based systems, voice interaction in video games, and complications with VUIs.

2.1 Research on Speech-Based Systems

Developing speech-based systems requires techniques, methodologies, and development tools that are capable of flexible and adaptive interaction, bearing in mind the need of different user groups and different environments [68]. In recent years, natural language processing (NLP) has become much more sophisticated and reliable [19]. Apart from technical development, interaction research tackled multitudes of novel voice interfaces, investigating how people use these devices and how they respond to different kinds of speech from computers [5, 9, 23, 41].

Speech-based systems have been evaluated for various purposes and professional fields. In the medical domain, Austerjost et al. presented a VUI for controlling laboratory instruments [7], while Miehle et al. presented a concept for voice assistants (VAs) as a support in surgical operating rooms [46]. For the purpose of teaching, Jung et al. [34] developed a voice-controlled educational game to teach children computer programming, concluding that their game led children to be more immersed in the game and understand the elements of programming with ease and confidence. Winkler et al. [73] compared groups who either used a human or a VA tutor when solving a problem. Their results indicated that groups

interacting with VA showed significantly higher task outcomes and higher degrees of collaboration quality compared to groups interacting with human tutors. Another prominent application area resides in entertainment. Zargham and Bonfert et al. [75] investigated voice interaction in a single-player VR game where they compared a version of the game in which the players could talk to multiple characters using natural language to a version where they verbally interact with a single character. The study showed that the participants preferred conversing with a group of interlocutors, found it more entertaining, and felt like being part of a team.

Although the functionality and ease of use of VUIs are frequently researched and enhanced, research suggests that the reliability of these systems is not more important than their attractiveness [74]. In a study by Lopatovska et al. [40], the authors explored user interactions with the popular VA Amazon Alexa. They report that people were still satisfied with the system even when Alexa did not produce desirable outcomes. Authors suggest that the UX might be more important to the users than the quality of the output.

One particular challenge with VUIs is that it can lead to unrealistic expectations from the system's intelligence, what it can do, and how well it can keep a natural and fluid conversation [43]. Users tend to test the capabilities of VUIs by asking different questions and in many cases, their expectations tend to exceed the agent's capabilities [41, 42]. This also applies to children. In a study by Lovato et al. on children's experience with Siri, authors found that children predominantly ask Siri personal questions, to get to know the agent and test its potential [41]. When users' initial expectations from such systems are not met, it can lead to disappointment and a generally negative experience [55].

Overall, a great deal of the design research is focused on narrow application areas and specific interface components. This in turn leads to the lack of more generalizable design guidelines [22]. In our work, we seek to advance the state of the art by exploring methodologies of recognition error handling.

2.2 Voice Interaction in Video Games

The intuitive nature of voice user interfaces allowed them to become an increasing trend, not only in assisting function within smart homes, phones or cars, but also for the advancement of mechanics within the entertainment industry. Although the rate of VUI studies has increased in recent years, research on voice interaction in games – those where voice control has a fundamental role in the game – is rather limited [20]. Using alternative means of interaction for games such as voice can not only expand the possibility space for novel in-game mechanics, but can also be especially important for users with disabilities, where traditional controls are not feasible [71]. Speech-controlled video games have also proven effective in enhancing speech therapy and facilitating remote treatment [1]. Other human modalities can be combined with speech to optimize players' performance and overcome the drawbacks of using only speech [50]. Nonetheless, there are still essential aspects and questions regarding voice interaction in games that have gone largely unexplored [5].

Voice interaction in video games is rather distinct from the other contexts. Research shows that in-game voice commands are associated with a sense of taking on a character in the game's world [4]. Allison et al. suggest that voice interactions which creates a conflict with the social world can impede the player's engagement with the in-game world [5]. Early research on voice interaction in digital games roots back to the 1970s, where *VoiceChess*, a game which could support standardized chess instructions using a speech recognition system, was developed [2, 59]. Since then, numerous video game titles have embraced the use of voice as input. In a successive study by Allison et al., the authors surveyed 449 video games and 22 audio games in which players use their voice to affect the game state [3]. They observed that academic research has focused on a narrow subset of design patterns, especially pronunciation, and recommend game designers to consider non-verbal forms, which have proven to provide enjoyable game experiences with fast and discrete input possibilities [32, 52, 64, 70].

Although there are plenty of examples of video games that use speech-based voice interaction, those which use non-verbal forms of voice input have been more successful. The reason behind the success of such games is that they avoid recognition errors entirely [3, 4]. However, due to the limited controls, these games are usually restricted to relatively simple mechanics. In this work, we simulate an environment that enables fast and reliable calculation of technically optimized actions so that gaps in recognition can be handled and the resulting experience investigated. To the best of our knowledge, no previous video game has used a similar technique, thus our approach of an anticipatory error handling method is original.

2.3 Complications with Voice Interaction

A large portion of research about voice interaction is concerned with speech recognition and its accuracy rates [3]. These systems are commonly trained with a large sample of voice data, connected with ontologies and knowledge graphs, in order to identify and understand users' commands and respond with a reasonable and satisfying answer [39]. Nevertheless, the given commands by the users can be fuzzy, personal, and complicated, resulting in the system not being able to understand them, which often leads to user frustration, disappointment, and dissatisfaction [10, 26, 42]. These issues are not likely to be overcome by soft- or hardware advancements in recognition alone. To conquer the difficulties inherent in processing the commands, users usually need to put more effort in formulating the command so that it is recognized by the system.

When interacting with a VUI, users typically speak differently than they would speak to a human. Many expect natural language not to be understood by such systems and adapt special communication strategies therefore. Reducing the talking pace, re-formulating command sentences and physically relocating themselves and/or the system are popular observable patterns when users are confronted with recognition errors [11]. Jentsch et al. observed that users took a considerable amount of time to formulate their prompts before commanding them to a VUI [11]. In their study, authors also witnessed that even when the users are not instructed to use keywords, they are still likely to restrict themselves to a set of words or commands when addressing a speech assistant. This has led users to refrain from speech-based systems to perform difficult tasks. In a study by Luger et al. [42], authors interviewed frequent users of conversational agents and found that the study participants did

not trust the system to do complex tasks – like writing emails or making phone calls – down to an apprehension that the system would not get the task done correctly. Authors also note that the interaction with the agent was generally considered as a secondary task.

On the other hand, when errors occur, the system should give an appropriate response. In her book about designing VUIs, Cathy Pearl suggests that, if the error handling is done well, it will not derail users, and you can get them back on the track and have them successfully complete a task [53]. If it's done poorly, not only the user will fail to complete a task, but they actually might refuse to use the system again. A study by Suhm et al. explored multimodal error correction methods that allows the user to correct the recognition errors in speech user interfaces [65]. The authors found that although users preferred speech as an input modality, if the accuracy of recognition was low, they learned to avoid it with experience. Vertanen et al. explored different techniques such as silence filtering to improve the recognition of spoken corrections when a system fails to recognize the command in the first try [69]. Their study showed that by combining multiple techniques, the percentage of correctly recognized spoken corrections increased by more than 30%. Bohus et al. subdivide speech recognition errors into two types of misunderstandings and non-understandings [13]. Misunderstandings are referred to those cases in which the system misinterprets the user's input, where in non-understanding events, the system fails to obtain any interpretation. In their study, the authors looked at ten non-understanding recovery strategies and compared their performance. Their results showed that advancing the conversation by ignoring the non-understanding and trying an alternative dialog plan performed best [13].

Although the technical aspects of VUIs have been largely investigated, researchers agree on the stance that the user side of speech interaction is relatively less explored [8, 22, 47, 49]. Above that, language barriers pose a further common problem with VUIs. A study by Pyae et al. showed that VUIs are easier to use, friendlier and potentially more useful for native English speakers than non-native speakers [58]. The complex and expensive process of implementing a reliable speech-based system, impels researchers in this field to often use a Wizard of Oz approach [36, 45].

Eventually, technical limits, unnatural assumptions, and lack of faith in the system's technological capabilities still make up the major reasons for users' reservations against using VUIs. To build upon the prior work regarding the error handling of speech systems, we came up with an approach to avoid unrecognized commands as well as repeating the command in order to correct it which could result in user frustration and ultimately abandoning the system entirely [53, 65]. In our approach, we focus on overcoming innate technical limits of speech recognition with anticipatory error handling and examine the impact of this intervention on the perceived intelligence, appraisal and usability of the system.

3 PROTOTYPE DESIGN

To evaluate our hypotheses, we designed and implemented "Listen, Sparky!", a speech-controlled arcade game. In this game, players are in control of the sheepdog "Sparky" who has to guide a sheep through restricted courses and keep away hazardous encounters. Using speech-controlled commands, players impersonate a shepherd that gives directions to his sheepdog. The game consists of eight levels. In every level, players have to safely navigate and return the sheep that escaped from a meadow, up to a designated goal location (gate).

The first four levels of the prototype served as a tutorial. In these, players were taught about the game controls and the commands to use. Every level would introduce one new command to the players, with the exception of the fourth level that would introduce two commands. The participants were able to access an overview of the available commands at any time in the game menu (see Figure 2). After going through the first two levels, a hostile wolf character was introduced that threatened the survival of the escorted sheep. If the sheep would get too close to the wolf, the level failed and had to be restarted. With increasing progression of the levels, the challenge of the game would similarly increase (see Figure 1). For instance, in the early levels, the wolf is standing still and does not move and the player has to simply avoid those areas of the game. In higher levels, the wolf would start moving or even chase the sheep to make the game more demanding for the player and enforce quick acting. At the end of each level, the game would display a screen indicating that the level was successfully completed while presenting performance feedback throughout a classic star rating system. This rating was given based on the number of commands used in that level and the time taken to finish it.

Level	*Is there a wolf?*	*Is the wolf moving?*	*Is the wolf chasing the sheep?*
1st	No	--	--
2nd	No	--	--
3rd	Yes	No	--
4th	Yes	No	--
5th	Yes	Yes	No
6th	Yes	Yes	No
7th	Yes	Yes	No
8th	Yes	Yes	Yes

Figure 1: The increasing complexity of the levels with the players' progression.

In order to start the speech recognition and have Sparky listen to the commands, players had to press and hold the spacebar. As long as the space bar was pressed, the default computer microphone was used to record the players' voice. If the space bar was released too fast, the system would not process that command. While holding the space bar, the player's voice input was recorded, processed and (if possible) interpreted as one of the following actions:

- "Walk towards": Sparky walks straight towards the sheep, navigating the sheep to the same direction.
- "Flank Left": Sparky flanks the sheep from the left side, navigating the sheep to the right side (relative to the fixed view angle of the participant).

Figure 2: Voice commands making up the core game controls, assessable anytime during gameplay.

- "Flank Right": Sparky flanks the sheep from the right side, navigating the sheep to the left side.
- "Back": Sparky goes back to the position where it began the level.
- "Bark at wolf": Sparky moves towards the wolf and barks. This results in paralyzing the wolf for some seconds and making it harmless to the sheep.

The system was able to handle multiple phrases per action. For instance, if players wanted to command Sparky to "flank right", they could also use phrases such as "go right!", "right side" or "move right". If a command was recognized by the voice recognition system, Sparky would execute the corresponding command. If no matching command was found, the system would consider that as a failed attempt. In such cases, the game would refer to the error handling system based on the respective experimental group. For every participant, the system recorded the error rates, which was the number of commands that were not recognized by the system throughout the session. In order to evaluate different error handling methods, we needed to ensure noticeable instances of recognition failure. To achieve this, both game versions were programmed to have a minimum overall error occurrence of 15% after the first ten commands. This means, if a player managed to get lower than the target error rate, the next request was intentionally misrecognized by the system (even if this turned out to only rarely occur). At the end of the session, all participants were told about the planted errors (minimum 15% overall errors). This was done last in the interviews to not influence any prior assessments.

The environment of the game and the game logic have been built with Unity 3D[1]. For speech recognition, the Google Cloud Speech-To-Text service[2] was used. The requests were directly sent to the Google services. We chose this service as it does not require any native library to run and makes the prototype compatible with any available platform. We created builds for Windows, Mac OS and Linux.

3.1 Anticipatory Error Handling

The anticipatory error handling was implemented to pick the best available option based on the current game state. In effect, if a command was not recognized, the game would perform a locally optimized action regarding obstacle avoidance and goal completion without letting the player know that the recognition failed. The game would first prioritize not getting eaten by the wolf (obstacle), and then would consider the action which would position the sheep closest to the gate (see figure 3). In the following section, we will give an example of how this procedure worked within the context of "Listen, Sparky".

4 EVALUATION

4.1 Study Design

We conducted a between-subjects design user study with (N = 34) participants to compare and evaluate our two conditions. In the control group, participants played a version that employed traditional

[1]https://unity3d.com/unity
[2]https://cloud.google.com/speech-to-text

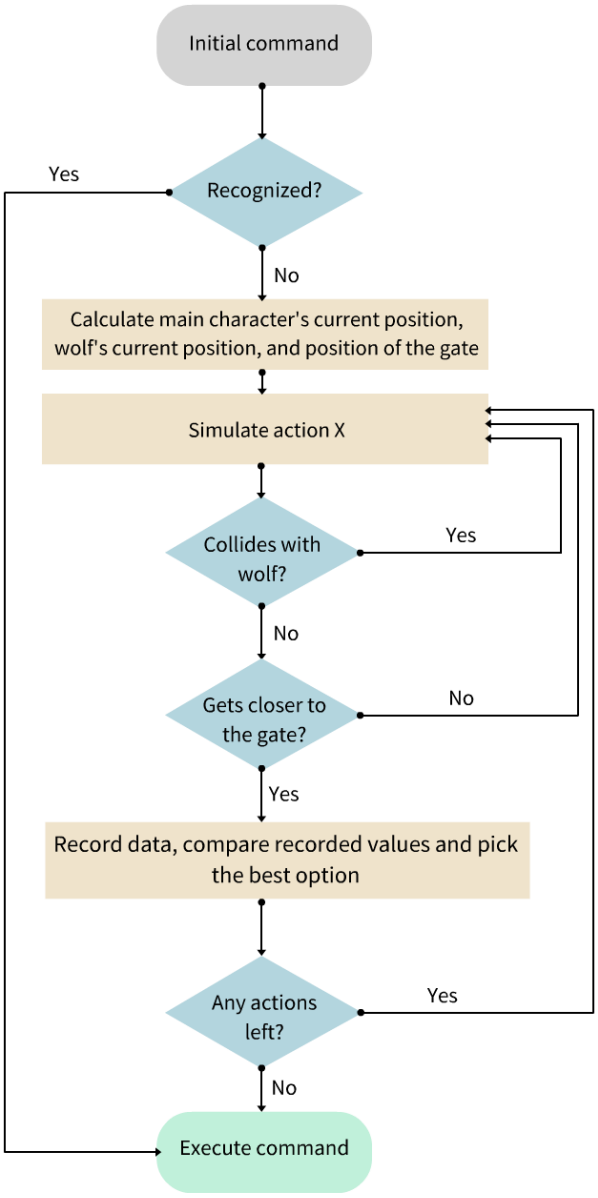

Figure 3: General process of the anticipatory error handling.

error handling, i.e. in the case of non-recognition, the character would not react but only indicate that the command was not recognized by displaying some question marks above its head (see Figure 4).

In the intervention group, players played a version that implemented anticipatory error handling, based on the underlying game state. For instance, considering the game situation in figure 5, the player commands sparky to "bark" but the intent was not recognized. The game would then refer to the error handling system that would then decide which action would be most optimal at

Figure 4: In the control group, when a command is not recognized, the game displays question marks over Sparky's head.

that moment so that the sheep can avoid getting eaten by the wolf and/or can get closer to the gate. The system then chooses "flank left" as the anticipated solution since it would have the best possible outcome where the sheep stays away from the wolf, and it gets close to the gate.

Among both conditions, levels, game environment, and mechanics remained equal, leaving the error handling method as the single manipulated variable. Group assignment was pseudo-randomized between two equally distributed groups. Participants were asked to play all eight levels of "Listen, Sparky" – yet, if they became stuck on a specific level after multiple tries, they were allowed to skip it. The execution took place on the subjects' own PC or laptop device. We sent an executable format of the game (build) to the participants prior to the session and made sure that every player had a functional microphone to use for the game.

4.2 Procedure

Every experimental session was held remotely via video calls. The experiment and interview were recorded acoustically and transcribed for later analysis. Furthermore, the experimenter noted verbal statements and in-game observations while providing assistance in cases of issues. Before starting the session, participants were briefly informed about the experiment procedure. Although the game contained an explanatory tutorial, the interview conductor would shortly explain the game and the controls. After the participants gave informed consent, they would share their screen with the experiment conductor. Participants would then play through the game in either one of the two conditions. They could also take a short break in between the levels. After finishing the game, participants completed the post-exposure questionnaires. At the end of the session, we held a short semi-structured interview with each participant. Each session took approximately 40 – 50 minutes, with an average of 18.4 minutes game-play time (SD = 5.16).

4.3 Measures

In order to evaluate our hypotheses and to understand how players experience the error handling in both conditions, we used standardized questionnaires to assess the player experience and the

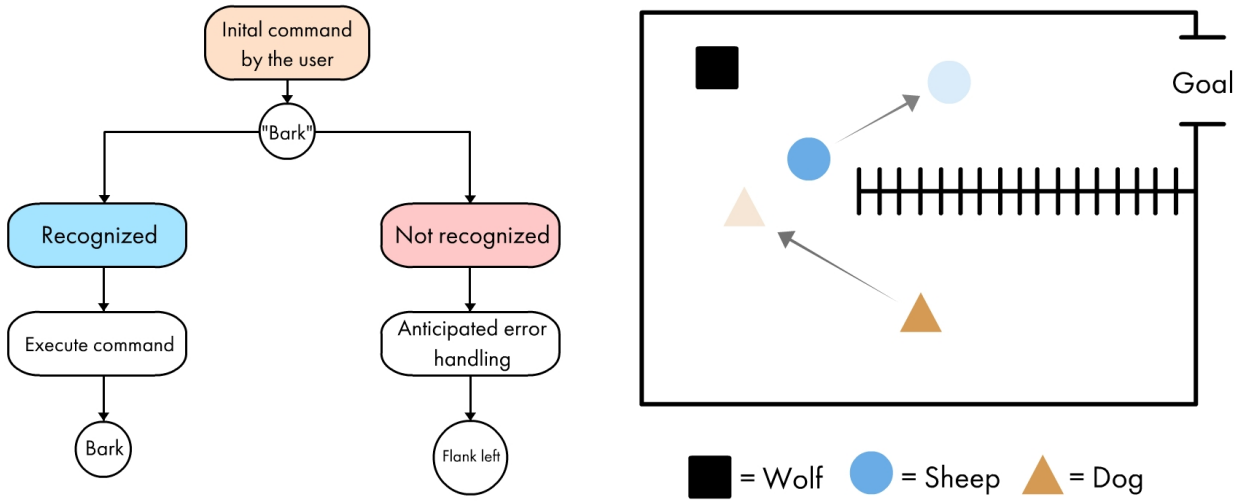

Figure 5: Displaying a specific game situation where the recognition fails and the system chooses to flank left as it would have the best possible outcome (right). The flowchart showing the process of the anticipatory error handling in the intervention group (left).

perceived usability of the system. Our post-exposure questionnaires included demographic questions, the System Usability Scale (SUS) [17], as well as the Player Experience of Need Satisfaction (PENS) [61] throughout the subscales of *Competency*, *Autonomy*, *Relatedness*, *Presence/immersion*, and *Intuitive controls*. Both questionnaires are validated and established measurement instruments. We chose SUS as it is a reliable tool for measuring usability of a system, which ensures high comparability. The PENS is also a validated questionnaire which determines the player experience. In our evaluation, we did not consider the sub-scale of *Relatedness* as it was not relevant to the scope of this study.

Additionally, we recorded a series of customized questions regarding their experience with the game. These were executed via 5-point Likert scales and concerned the extent with which Sparky behaved as the participant expected him to do so, Sparky's perceived intelligence and the overall experience with the game. Above that, players were asked to estimate the approximate number of commands that were not recognized, and to explicate what Sparky did when the commands were not recognized by the system. For all statistical tests, we applied an alpha level of .05. We concluded the session with a brief, semi-structured interview to further evaluate qualitative aspects of player experience, usability, and individual preferences for both conditions [72]. The interview recordings were systematically examined. For this, two researchers agreed on a coding system that was generated from a random selection of ten interviews. Subsequently, all recordings were analyzed, coded along this categorization, and summarized. Additionally, we collected insightful and unique statements.

4.4 Participants

A quota sampling approach was used to recruit participants for this study in which the selection was based on mailing lists, social networks, word-of-mouth and gaming forums. Participation was voluntary and uncompensated. ($N = 34$) people participated in the experiment. In the control group, 17 participants (5 self-identified as female, 12 as male) between 22 and 43 years of age ($M = 29.64$, $SD = 5.42$) played a version of the game with traditional error handling. In the intervention group, 17 players (5 self-identified as female, 12 as male) which were mutually excluded from the first group, between 22 and 38 years of age ($M = 27.7$, $SD = 4.87$) played a version that implemented anticipatory error handling. 85% of our participants had previous experience with voice assistants (18 rarely, 11 often). Only 17% of the participants have previously played a voice-controlled video game. We conducted the experiment in English with international participants. The sample consisted of two native English speakers and the rest were fluent non-native English speakers.

5 RESULTS

In order to identify possible differences between both conditions, we applied Mann-Whitney U Test as well as qualitative content analysis towards our issued research questions.

Four participants did not fill in an item within the Autonomy subscale of PENS. These missing values were imputed by the average value. In our study, we focused on the four sub-scales of *Competency*, *Autonomy*, *Presence/Immersion*, and *Intuitive Controls* (cf. Figure 6). Consequential, we found a significant effect for *Intuitive Controls* in favor for the intervention group ($M = 5.96$, $SD = 1.29$), compared to the control group ($M = 4.7$, $SD = 1.98$), $U = 84.5$, $p = .040$, displaying a medium effect ($d_{Cohen} = 0.75$) [24]. In contrast, *Competency*, *Autonomy*, and *Presence/Immersion* did not show significant differences between the two conditions ($p > .05$).

Regarding usability, SUS scores reached an average of 63.23 ($SD = 20.47$) within the control group, whereas the intervention

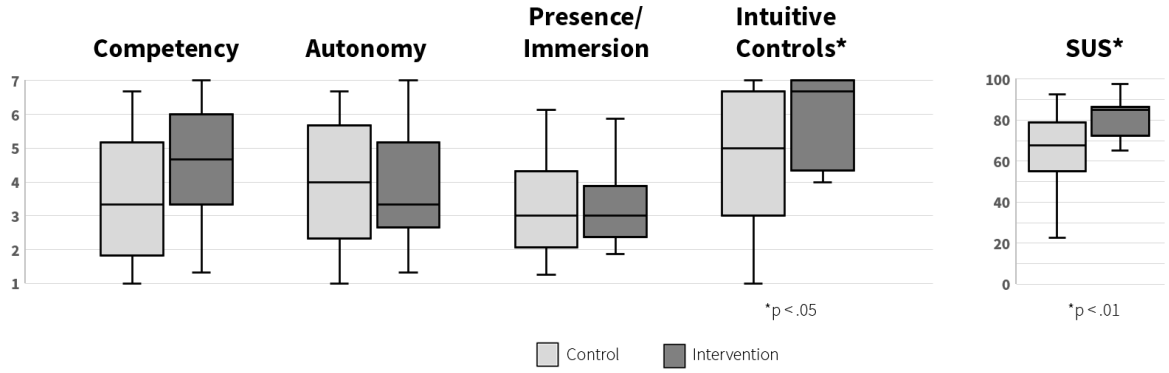

Figure 6: Boxplot indicating significant results from PENS-subscales and SUS between control and intervention group. Includes median (–), standard deviation (box) and range (whiskers).

group resulted in 80.88 (SD = 8.96). The subsequent Mann-Whitney U Test indicates that anticipatory error handling outperformed the control group significantly in terms of usability (U = 65.5, p = .0069, cf. Figure 6), revealing a large effect between conditions (d_{Cohen} = 1.055). Any SUS score higher than 68 would be considered above average, and anything lower is below average [16]. Therefore, the results indicate an above average usability score for the intervention conditions and a below average one for the control condition. For the overall game experience, players of the control group rated it as 3.411 (SD = 1.18) on average, not significantly different from the intervention group (M =3.889, SD = 0.93; U = 111, p = .254). Assessing to what extent Sparky followed the users' expectations, no significant differences between the control (M = 3.12, SD = 0.99) and intervention group (M = 3.24, SD = 0.90) could be found (U = 134.5, p = .74). Similarly, no significant effect on Sparky's perceived intelligence emerged (U = 134, p = .72), with an average of 3.06 (SD = 0.97) under the control condition, and 2.94 (SD = 1.14) within the intervention group.

Overall, the participants in the intervention group had a mean error rate of 42.94% (SD = 18.41), while the control group resulted in 33.53% (SD = 11.31) errors on average. This showed no significant differences between the two conditions in terms of error rates (U = 98, p = .114).

However, when participants were asked to write down the approximate number of commands that were not recognized by the system, the mean number of perceived errors in the control group resulted in 34.863 (SD = 36.882) which is significantly higher (U = 63.5, p = .0056) than that of the intervention group (M = 6.438, SD = 5.501), revealing a large effect (d_{Cohen} = 1.09).

We also asked the participants to explain Sparky's behavior in cases where commands were not correctly recognized by the system. In the intervention group, 59% believed it did something wrong, 23% said it did something random, 12% said it always understood the commands, and 6% thought, it helped to perform the right action. Among the participants under the control condition, 76% said Sparky did not react when the command was not recognized, 12% said it did something wrong, one participant (6%) said it did something random and another stated that the commands were always recognized.

5.1 Qualitative Results

Interpreting the post-exposure interview sessions, qualitative insights could be extracted with respect to the different error handling methodologies and the overall game experience itself.

5.1.1 Overall Impressions. Participants generally enjoyed playing the game and attributed it as entertaining. Some (21%) even asked for repeating the game levels to get more playtime. In both groups, players liked the idea of playing a speech-based video game in general and found it novel. Many players (15 of 34=44%) stated that they perceived the game's controls as intuitive and several (41%) mentioned that they especially liked the game's aesthetics. Four participants fancied the background music and sound effects used in the game while one user found it distracting. Three participants gave suggestions regarding the use of speech-based games in serious contexts such as teaching and therapy. One participant specifically mentioned that speech-based games such as this game could be an interesting medium to teach foreign languages to children. Two participants stated that they felt like an actual shepherd in control of a dog. One participant said, "I had a feeling of control because the dog behaved as I intended. I was in command of the dog."

5.1.2 Progressive Enhancement. In both groups, participants mentioned that they got better at controlling Sparky after some playing time. One participant stated: "I felt that I learned how to speak for the game to understand me". However, some (10 of 34=29%) believed that with their improvements, the game's challenges also got more complex. Eight players (29%) stated that they specifically enjoyed the progressive enhancement of the game's difficulty. One user suggested using different difficulty levels, where the recognition gets worse when you increase the difficulty.

5.1.3 Voice Command For Game Control. During the sessions, we observed that all participants looked at the commands list more than once. Even though the controls were rather limited, participants were struggling to memorize them all. We also noticed that our participants would look at this list to use the exact phrase suggested in that screen. Although, the recognition system was able to handle different styles of commands in the same context that were likely

to be given by participants, and thus not limited to the particular commands from the tutorial. This was observed by several players. One of the participants stated: "The commands were intuitive. I did not use exactly the game's commands, and it still worked. I liked that". On the other hand, players wished for fewer restrictions regarding the commands for the game controls. One player said, "I'd expect all the normal replacement phrases to work as well". Some (12%) participants shared an opinion that more controls would be helpful, e.g. one participant stated that "it would be nice to have a command that repeats the previous one". One participant said that single-word commands would be better for such games. Few (9%) believed that using phrases felt more natural and interesting. On multiple occasions, we witnessed that the participants' voices were raised, or they spoke faster when they were under time pressure and had to make quick decisions, which likely led to a higher error rate.

5.1.4 Recognition and Error Handling. Both groups equally (five participants in each group) reported the disliking of the occurrence of voice recognition malfunctioning, as well as the delay between the command and execution. Two participants (both none-native English speakers) expressed their struggle with the recognition due to their accent and mentioned that it would have been nice if the system could learn their voice and accent. One player in particular found it entertaining that Sparky could not understand all the commands: "It felt more realistic this way". Three participants (two from the intervention group, one from the control group) believed that 100% of their commands were recognized by the system, although none of the participants had a smaller error rate than 17%.

Seven participants in the intervention group mentioned that they sometimes found the behavior of Sparky unexpected. Only one player in the control condition mentioned something similar. During the interviews, we revealed both conditions and their difference in error handling to the participants. Four of them mentioned that they would prefer to have anticipatory error handling as an optional feature that they could activate in the game's settings. One participant stated, "When the recognition is not working, that means there is a problem. If I don't see the errors, I don't see the problem. So I think the errors should be seen to acknowledge the problem and improve the recognition". Another mentioned "I would personally choose this version [repetition-based] as I want to have full control of the game." Multiple participants (15%) of the intervention group shared the opinion that they like that the game's flow is not being disturbed by recognition errors. One of them stated: "I really like the idea of this game since it does not disturb the flow when there is an issue with the recognition technology". One participant said, "I would prefer that the game performs an action randomly. That way, it makes the game more exciting and challenging".

6 DISCUSSION

This evaluation aimed at exploring the impact of recognition error handling techniques on the user experience by contrasting traditional to anticipated handling within a speech-controlled video game. Overall, users' feedback about "Listen, Sparky!" were rather positive and supporting. Players in both conditions generally enjoyed playing our voice-controlled game. During the experiment, participants asked for repeating the levels even after successfully finishing that level. They also wanted to continue playing after the experiment was done. Three of our participants specifically pointed out that controlling Sparky with voice made them feel more immersed as they felt 'like an actual shepherd', supporting the findings by Allison et al. [4], that the player's in-game voice commands can be associated with a feeling of taking on a character in the game's world. Moreover, we witnessed a significant difference in terms of intuitive control between the two conditions. This can imply that implementing optimal error handling can lead to a higher perceived intuitiveness of a system.

Many players expressed their struggle with the recognition of their commands, especially in the beginning of the game. We observed that participants improved in understanding how the recognition system works after spending some time in the game. They learned how to formulate their commands and to speak clearly in order to be recognized by the system. Additionally, they also developed their ability to play the game by adopting the game mechanics over the various levels. Furthermore, we saw that many participants looked at the controls screen multiple times during the game to use the exact phrases suggested in that screen, even though the recognition system was able to handle different types of commands for the same action. This was inline with the previous work by Jentsch et al. [11] who also mentioned that the users adapt special communication strategies to speak to the system.

Additionally, we observed that players often perceived time pressure, leading to more complications with command recognition. This was mainly due to the change in the talking pace and fast decisions, which at times led to unclear and incorrect inquiries. We also recorded a higher error rate for non-native speakers. This led to more frustration for these players during the game, aligning with the results of the study by Pyae et al [58].

Eventually, we interpreted the results of this experiment to provide answers to the following comprehensive questions:

RQ1: Does performing a locally optimized game action in times of misrecognition lead to a measurably improved usability in a speech-based video game?
RQ2: What are the effects on player experience in terms of competency, autonomy, presence, and intuitive control, if error handling mechanisms decide for unintended actions?

Regarding **RQ1**, results indicate a significantly higher usability, as well as higher ratings of intuitive control for the version employing anticipatory error handling. Yet, qualitative statements underline that this increase of usability is mainly due to the cases where the error handling actually followed the user's intention, which was not always the case, even when deciding for the technically optimized solution. In cases of mismatch, participants perceived it as a different kind of error, even if the performed action was the technically optimized choice. As soon as doubts about the system were raised, the learning curve of the users was also impacted. Thus, we argue that error handling can improve the user experience of speech-based games, though the major objective of the handling technique should not approximate technically optimized decisions, but individually tailored predictions. Supplementary to the usability analysis, quantitative findings of the recorded error observations

confirm the former results: Although participants of the intervention group committed more errors on average, they in fact reported a significantly lower amount of perceived errors, compared to the control group. In effect, we accept our first hypothesis:
H1: Participants will observe a lower number of recognition errors in case of anticipatory error handling.
Even though this was partially caused by the fact that in the intervention group, a certain number of unrecognized commands by the users were in fact the optimized action, therefore no recognition failure was perceived. Nonetheless, even when a misrecognition was handled by an (optimized) action that deviated from the intended command, users were still less likely to detect this intervention.

Furthermore, based on the results of the PENS questionnaire, we accept $H2_d$, while rejecting $H2_a$, $H2_b$, and $H2_c$. Concerning **RQ2**, we observed differences between both groups and interpreted users' reactions and responses to error handling that conflicted with their original intention. Players of the intervention group were repeatedly confused by Sparky acting against their original intention, resulting in a misleading learning experience that impaired in-game progress and proficiency attainment. Since the control group was not affected by automatically handled actions, this issue did only occur in the former condition. Even if quantitative insights suggest a higher usability through the anticipatory error handling intervention, qualitative statements reflect the dissatisfaction in situations where the handling deviates from the user's intention. Above that, since correctly handled errors were not perceived as errors in the first place, participants rated the intervention version as not more intelligent than the without handling.

After we revealed both conditions to our participants, we witnessed a mixture of opinions regarding the different error handling methodologies. Some were in favor of the anticipatory error handling as it helped to keep the flow of the game. Some didn't like it as they believed it hides the problem rather than solving it. One participant also proposed performing a random action rather than an optimized one to make the game more challenging and add an element of surprise. Considering all the differences in the opinions, one can assume that the optimal solution could differ from one player to another. Game developers can consider equipping different methods as optional features of the game, where the players can choose their desired methods based on their own preferences.

In our study, we used a limited set of commands to focus on the error handling methodologies. In cases of larger command sets, the system can eliminate those commands which are out of the current context and between those left, choose the most suitable action based on the situation and previous user behavior. Depending on the application, one can take the action with the highest probability or present the users with a number of top possible actions to choose from. Previous research on repair strategies with chatbots has shown that system-repair where the chatbot provides possible options to users was arguably favored by the users as it required less effort from the user to repeat their inquiry [6]. This can likely be enhanced with machine learning techniques and user models.

Predicting user's intent to improve usability and user experience is not a new topic. In terms of conversational agents for instance, there has been extensive research on predicting user intents and deciding for an appropriate repair strategy in case of a conversation breakdown [6, 37, 62]. In the context of video games, however, this area is still under investigated. There have been attempts to employ deep learning to provide adequate models of individual player behavior with high accuracy [54], or opponent modeling to predict different strategy patterns of opponents [33]. However, to the best of our knowledge, this is the first work aiming at alternative strategies of error handling in times of command recognition failures in voice-controlled video games. In this study, we witnessed that players did not necessarily favor the cases that the anticipatory error handling was used if the action did not match their initial intent. One could assume that having full control over the game and perceiving a feeling of agency could be rather preferred, even if their actions are not the optimal ones towards level completion. Although, repeating the game actions when they are not recognized was even more frustrating. Therefore, more effort should be put on understanding the user's initial actions rather than finding the optimal action. Nonetheless, more aspects such as player types, mood, and game genre's need to be investigated in order to gain a deeper insight in this regard.

Based on the interpretation of the results regarding both research questions, we conclude with the following implications: Error handling can significantly improve the usability of a speech-controlled video game and aid in bridging the technological gap of speech recognition. Yet, ideal error handling should model (and predict) the individual user's intention, be equipped with an internal likelihood estimation whether the handled decision is appropriate or follow similar methods to ensure user satisfaction. Otherwise, false handling can impair both the experience as well as the learning progress and raise doubts about error handling in general. This work successfully demonstrated a first approach of anticipatory error handling, but these "optimal decisions" from a heuristic can still deviate from the user's intention. Future work will extend this by approximating the users' intention even further (e.g. creating user models).

6.1 Limitations and Future Work

While the findings of this study present significant steps forward in exploring recognition error handling methodologies in speech-based games, there are still some limitations that should be addressed. In this work, we investigated anticipatory error handling in a speech-based video game. Although the broader insights of this evaluation can apply to the use and error handling of VUIs in general, in future work, these methods could be transferred and evaluated in other domains such as navigation, medicine, education, and smart homes, to explore conversationally more complex settings.

The anticipatory error handling used in this study could also raise certain ethical concerns in terms of misleading the user into thinking that no error has occurred. Although this may not be a big concern in the context of most video games and the approach may help players with speech recognition, developers implementing this method should transparently make information about the history of errors and the commands that lead to the error handling available to the users. Furthermore, it may also be necessary that the method is explained to the players.

Another concern that could be raised is that certain players may abuse such features by purposefully giving unclear commands,

being certain that the system would perform the optimal action. Although we did not observe such behavior during the experiment, we recommend designers and developers to consider methods to prevent players from misusing this feature, for instance, by observing odd behavior from the player such as repeated unrecognized commands.

During the experiment, we noticed that some participants had difficulties learning the game controls and game mechanics. For future studies, we recommend longer tutorials as well as gaming sessions to counter influences on individual learning rate. Apart from this, differences in player types and players' current emotional and social states could lead to different experiences, which should be incorporated and reflected in further studies. The implemented voice recognition system for the game has not been trained with data from non-native English speakers, yet the majority of participants fell under this condition. The recognition with those who spoke a strong accent was therefore not optimal and could have been improved by training the system differently. Although our game controls were limited to a predefined set of commands, this helped us to have a structured procedure with high comparability [57]. The focus of this work was to study occurrences of recognition failures and the subsequent handling and not to engineer a solution for a large-scale complex system. In order to yield scalable insights for broader application fields and cover large command vocabularies, future studies will expand the scope of the potential actions.

This was a first exploration on anticipatory error-handling in video games. Our experiment sample consisted of ($N = 34$) mostly male users (70.6%). An influence of such bias on the results can not be excluded. Moreover, future studies could validate our findings by investigating a wider population.

The positive feedback and enthusiasm towards our game can be partially affiliated by the unconventionality of speech-based video games in general. The demographics data as well as the perceived novelty of the game by the participants shows that voice-controlled games are still an unfamiliar category. In this paper, we demonstrated that utilizing speech-based interaction in games can help to increase inclusion and as well as immersion as you are actively communicating with in-game characters instead of just pressing buttons. We further encourage researchers in this field to investigate the area.

7 CONCLUSION

In this paper, we investigated anticipatory error handling for a speech recognition system and explored its potentials and challenges. We designed a voice-controlled video game called "Listen, Sparky!" to evaluate our concept. In a between-subjects design study, we compared our anticipatory error handling model to a traditional repetition-based version. Our results showed that implementing anticipatory error handling can improve the usability of a system, if it follows the intention of the user. Otherwise, it can impair the user experience, even when making technically optimized decisions. Ideal error handling should therefore model the individual user's intention, be equipped with an internal likelihood estimation whether the handled decision is appropriate, or follow similar methods to ensure user satisfaction. Our findings contribute useful insights for researchers and developers on how to address, display and handle recognition errors in speech-based video games and the greater application field of voice user interfaces.

ACKNOWLEDGMENTS

This work was partially funded by Klaus Tschira Foundation, by the FET-Open Project 951846 "MUHAI – Meaning and Understanding for Human-centric AI" funded by the EU program Horizon 2020, as well as the German Research Foundation DFG as part of Collaborative Research Center (Sonderforschungsbereich) 1320 "EASE – Everyday Activity Science and Engineering", University of Bremen (http://www.ease-crc.org/) conducted in subproject H02.

REFERENCES

[1] Beena Ahmed, Penelope Monroe, Adam Hair, Chek Tien Tan, Ricardo Gutierrez-Osuna, and Kirrie J Ballard. 2018. Speech-driven mobile games for speech therapy: User experiences and feasibility. *International journal of speech-language pathology* 20, 6 (2018), 644–658.

[2] Fraser Allison, Marcus Carter, and Martin Gibbs. 2017. Word Play: A History of Voice Interaction in Digital Games. *Games and Culture* 15, 2 (2017), 91 – 113. https://doi.org/10.1177/1555412017746305

[3] Fraser Allison, Marcus Carter, Martin Gibbs, and Wally Smith. 2018. Design Patterns for Voice Interaction in Games. In *Proceedings of the 2018 Annual Symposium on Computer-Human Interaction in Play* (Melbourne, VIC, Australia) *(CHI PLAY '18)*. Association for Computing Machinery, New York, NY, USA, 5–17. https://doi.org/10.1145/3242671.3242712

[4] Fraser Allison, Joshua Newn, Wally Smith, Marcus Carter, and Martin Gibbs. 2019. Frame Analysis of Voice Interaction Gameplay. 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/10.1145/3290605.3300623

[5] Fraser John Allison. 2020. *Voice interaction game design and gameplay*. Ph.D. Dissertation. University of Melbourne.

[6] Zahra Ashktorab, Mohit Jain, Q. Vera Liao, and Justin D. Weisz. 2019. Resilient Chatbots: Repair Strategy Preferences for Conversational Breakdowns. 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/10.1145/3290605.3300484

[7] Jonas Austerjost, Marc Porr, Noah Riedel, Dominik Geier, Thomas Becker, Thomas Scheper, Daniel Marquard, Patrick Lindner, and Sascha Beutel. 2018. Introducing a virtual assistant to the lab: A voice user interface for the intuitive control of laboratory instruments. *SLAS TECHNOLOGY: Translating Life Sciences Innovation* 23, 5 (2018), 476–482.

[8] Matthew P Aylett, Per Ola Kristensson, Steve Whittaker, and Yolanda Vazquez-Alvarez. 2014. None of a CHInd: relationship counselling for HCI and speech technology. In *CHI'14 Extended Abstracts on Human Factors in Computing Systems*. ACM, New York, NY, USA, 749–760.

[9] Frank Bentley, Chris Luvogt, Max Silverman, Rushani Wirasinghe, Brooke White, and Danielle Lottridge. 2018. Understanding the Long-Term Use of Smart Speaker Assistants. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 3, Article 91 (Sept. 2018), 24 pages. https://doi.org/10.1145/3264901

[10] Frank Bentley, Chris Luvogt, Max Silverman, Rushani Wirasinghe, Brooke White, and Danielle Lottridge. 2018. Understanding the long-term use of smart speaker assistants. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 3 (2018), 1–24.

[11] Maresa Biermann, Evelyn Schweiger, and Martin Jentsch. 2019. Talking to Stupid?!? Improving Voice User Interfaces. https://doi.org/10.18420/muc2019-up-0253

[12] Dan Bohus and Alexander I Rudnicky. 2005. Constructing accurate beliefs in spoken dialog systems. In *IEEE Workshop on Automatic Speech Recognition and Understanding, 2005.* IEEE, IEE, New York, NY, USA, 272–277.

[13] Dan Bohus and Alexander I Rudnicky. 2008. Sorry, I Didn't Catch That! In *Recent trends in discourse and dialogue*. Springer, New York, NY, USA, 123–154.

[14] Richard A Bolt. 1980. "Put-that-there" Voice and gesture at the graphics interface. In *Proceedings of the 7th annual conference on Computer graphics and interactive techniques*. Association for Computing Machinery, New York, NY, USA, 262–270.

[15] Cynthia Breazeal. 2003. Toward sociable robots. *Robotics and autonomous systems* 42, 3-4 (2003), 167–175.

[16] John Brooke. 2013. SUS: a retrospective. *Journal of usability studies* 8, 2 (2013), 29–40.

[17] John Brooke et al. 1996. SUS-A quick and dirty usability scale. *Usability evaluation in industry* 189, 194 (1996), 4–7.

[18] Raluca Budiu and Page Laubheimer. 2018. Intelligent assistants have poor usability: A user study of Alexa, Google assistant, and Siri.

[19] Erik Cambria and Bebo White. 2014. Jumping NLP curves: A review of natural language processing research. *IEEE Computational intelligence magazine* 9, 2 (2014), 48–57.

[20] Marcus Carter, Fraser Allison, John Downs, and Martin Gibbs. 2015. Player Identity Dissonance and Voice Interaction in Games. In *Proceedings of the 2015 Annual Symposium on Computer-Human Interaction in Play* (London, United Kingdom) *(CHI PLAY '15)*. Association for Computing Machinery, New York, NY, USA, 265–269. https://doi.org/10.1145/2793107.2793144

[21] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. 2016. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, IEEE, New York, NY, USA, 4960–4964.

[22] Leigh Clark, Philip Doyle, Diego Garaialde, Emer Gilmartin, Stephan Schlögl, Jens Edlund, Matthew Aylett, João Cabral, Cosmin Munteanu, Justin Edwards, et al. 2019. The State of Speech in HCI: Trends, Themes and Challenges. *Interacting with Computers* 31, 4 (2019), 349–371.

[23] Leigh Clark, Nadia Pantidi, Orla Cooney, Philip Doyle, Diego Garaialde, Justin Edwards, Brendan Spillane, Emer Gilmartin, Christine Murad, Cosmin Munteanu, and et al. 2019. What Makes a Good Conversation? Challenges in Designing Truly Conversational Agents. 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, Article 475, 12 pages. https://doi.org/10.1145/3290605.3300705

[24] Jacob Cohen. 1988. *Statistical Power Analysis for the Behavioral Sciences* (2nd ed.). Erlbaum, Hillsdale, NJ.

[25] Erica Cooper, Alison Chang, Yocheved Levitan, and Julia Hirschberg. 2016. Data Selection and Adaptation for Naturalness in HMM-Based Speech Synthesis. In *Proc. Interspeech 2016*. ISCA, France, 357–361. https://doi.org/10.21437/Interspeech.2016-502

[26] Benjamin R Cowan, Nadia Pantidi, David Coyle, Kellie Morrissey, Peter Clarke, Sara Al-Shehri, David Earley, and Natasha Bandeira. 2017. "What can i help you with?" infrequent users' experiences of intelligent personal assistants. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services*. ACM, New York, NY, USA, 1–12.

[27] Crytek. 2013. *Ryse: Son of Rome*. Game [XBox One]. Microsoft Studios, Redmond, Washington, U.S.

[28] Mihaly Csikszentmihalyi. 1990. *Flow: The psychology of optimal experience*. Vol. 1990. Harper & Row, New York, NY, USA.

[29] Steven Dow, Manish Mehta, Ellie Harmon, Blair MacIntyre, and Michael Mateas. 2007. Presence and engagement in an interactive drama. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, New York, NY, USA, 1475–1484.

[30] EA Sports. 2013. *Fifa 14*. Game [XBox One]. Microsoft Studios, Redwood City, California, U.S.

[31] Electronic Arts. 2012. *Mass Effect 3*. Game [XBox 360]. Electronic Arts, Redwood City, California, U.S.

[32] Susumu Harada, Jacob O Wobbrock, and James A Landay. 2011. Voice games: investigation into the use of non-speech voice input for making computer games more accessible. In *IFIP Conference on Human-Computer Interaction*. Springer, Springer, New York, NY, USA, 11–29.

[33] He He, Jordan Boyd-Graber, Kevin Kwok, and Hal Daumé III. 2016. Opponent modeling in deep reinforcement learning. In *International conference on machine learning*. PMLR, PMLR, New York, New York, USA, 1804–1813.

[34] Hyunhoon Jung, Hee Jae Kim, Seongeun So, Jinjoong Kim, and Changhoon Oh. 2019. TurtleTalk: an educational programming game for children with voice user interface. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1–6.

[35] Jesper Juul. 2007. Without a goal: on open and expressive games. , 191–203 pages.

[36] John F Kelley. 1983. An empirical methodology for writing user-friendly natural language computer applications. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 193–196.

[37] Knut Kvale, Olav Alexander Sell, Stig Hodnebrog, and Asbjørn Følstad. 2019. Improving Conversations: Lessons Learnt from Manual Analysis of Chatbot Dialogues. In *International Workshop on Chatbot Research and Design*. Springer, Springer, New York, NY, USA, 187–200.

[38] Kwan Min Lee, Wei Peng, Seung-A Jin, and Chang Yan. 2006. Can robots manifest personality?: An empirical test of personality recognition, social responses, and social presence in human–robot interaction. *Journal of communication* 56, 4 (2006), 754–772.

[39] Toby Jia-Jun Li, Igor Labutov, Brad A Myers, Amos Azaria, Alexander I Rudnicky, and Tom M Mitchell. 2018. An end user development approach for failure handling in goal-oriented conversational agents.

[40] Irene Lopatovska, Katrina Rink, Ian Knight, Kieran Raines, Kevin Cosenza, Harriet Williams, Perachya Sorsche, David Hirsch, Qi Li, and Adrianna Martinez. 2019. Talk to me: Exploring user interactions with the Amazon Alexa. *Journal of Librarianship and Information Science* 51, 4 (2019), 984–997. https://doi.org/10.1177/0961000618759414 arXiv:https://doi.org/10.1177/0961000618759414

[41] Silvia Lovato and Anne Marie Piper. 2015. "Siri, is This You?": Understanding Young Children's Interactions with Voice Input Systems. In *Proceedings of the 14th International Conference on Interaction Design and Children* (Boston, Massachusetts) *(IDC '15)*. ACM, New York, NY, USA, 335–338. https://doi.org/10.1145/2771839.2771910

[42] Ewa Luger and Abigail Sellen. 2016. "Like Having a Really Bad PA": The Gulf Between User Expectation and Experience of Conversational Agents. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. ACM, New York, NY, USA, 5286–5297. https://doi.org/10.1145/2858036.2858288

[43] Michal Luria, Samantha Reig, Xiang Zhi Tan, Aaron Steinfeld, Jodi Forlizzi, and John Zimmerman. 2019. Re-Embodiment and Co-Embodiment: Exploration of Social Presence for Robots and Conversational Agents. In *Proceedings of the 2019 on Designing Interactive Systems Conference* (San Diego, CA, USA) *(DIS '19)*. Association for Computing Machinery, New York, NY, USA, 633–644. https://doi.org/10.1145/3322276.3322340

[44] Michal Luria, Rebecca Zheng, Bennett Huffman, Shuangni Huang, John Zimmerman, and Jodi Forlizzi. 2020. Social Boundaries for Personal Agents in the Interpersonal Space of the Home. 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/10.1145/3313831.3376311

[45] David Maulsby, Saul Greenberg, and Richard Mander. 1993. Prototyping an intelligent agent through Wizard of Oz. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*. Association for Computing Machinery, New York, NY, USA, 277–284.

[46] Juliana Miehle, Daniel Ostler, Nadine Gerstenlauer, and Wolfgang Minker. 2017. The next step: intelligent digital assistance for clinical operating rooms. *Innovative surgical sciences* 2, 3 (2017), 159–161.

[47] Cosmin Munteanu, Matt Jones, Sharon Oviatt, Stephen Brewster, Gerald Penn, Steve Whittaker, Nitendra Rajput, and Amit Nanavati. 2013. We need to talk: HCI and the delicate topic of spoken language interaction. In *CHI'13 Extended Abstracts on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 2459–2464.

[48] Christine Murad and Cosmin Munteanu. 2019. "I don't know what you're talking about, HALexa" the case for voice user interface guidelines. In *Proceedings of the 1st International Conference on Conversational User Interfaces*. Association for Computing Machinery, New York, NY, USA, 1–3.

[49] Christine Murad, Cosmin Munteanu, Benjamin R Cowan, and Leigh Clark. 2019. Revolution or Evolution? Speech Interaction and HCI Design Guidelines. *IEEE Pervasive Computing* 18, 2 (2019), 33–45.

[50] Moyen Mohammad Mustaquim. 2013. Automatic speech recognition-an approach for designing inclusive games. *Multimedia tools and applications* 66, 1 (2013), 131–146.

[51] Aasish Pappu and Alexander Rudnicky. 2014. Knowledge acquisition strategies for goal-oriented dialog systems. In *Proceedings of the 15th annual meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*. Association for Computational Linguistics, Philadelphia, USA, 194–198.

[52] Jim R Parker and John Heerema. 2008. Audio interaction in computer mediated games.

[53] Cathy Pearl. 2016. *Designing voice user interfaces: principles of conversational experiences*. " O'Reilly Media, Inc.", Sebastopol, California.

[54] Johannes Pfau, Jan David Smeddinck, and Rainer Malaka. 2018. Towards deep player behavior models in mmorpgs. In *Proceedings of the 2018 Annual Symposium on Computer-Human Interaction in Play*. ACM, New York, NY, USA, 381–392.

[55] Martin Porcheron, Joel E. Fischer, Stuart Reeves, and Sarah Sharples. 2018. Voice Interfaces in Everyday Life. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. ACM, New York, NY, USA, Article 640, 12 pages. https://doi.org/10.1145/3173574.3174214

[56] Martin Porcheron, Joel E. Fischer, and Sarah Sharples. 2017. "Do Animals Have Accents?": Talking with Agents in Multi-Party Conversation. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing* (Portland, Oregon, USA) *(CSCW '17)*. ACM, New York, NY, USA, 207–219. https://doi.org/10.1145/2998181.2998298

[57] Robert Porzel and Manja Baudis. 2004. The Tao of CHI: Towards Effective Human-Computer Interaction. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*. Association for Computational Linguistics, Boston, Massachusetts, USA, 209–216. https://www.aclweb.org/anthology/N04-1027

[58] Aung Pyae and Paul Scifleet. 2018. Investigating differences between native English and non-native English speakers in interacting with a voice user interface: A case of Google Home. In *Proceedings of the 30th Australian Conference on Computer-Human Interaction*. ACM, New York, NY, USA, 548–553.

[59] D Reddy, Lee Erman, and R Neely. 1973. A model and a system for machine recognition of speech. *IEEE Transactions on Audio and Electroacoustics* 21, 3 (1973), 229–238.

[60] Mihai Rotaru, Diane J Litman, and Katherine Forbes-Riley. 2005. Interactions between speech recognition problems and user emotions.

[61] Richard M Ryan, C Scott Rigby, and Andrew Przybylski. 2006. The motivational pull of video games: A self-determination theory approach. *Motivation and emotion* 30, 4 (2006), 344–360.

[62] Amir Shevat. 2017. *Designing bots: Creating conversational experiences.* " O'Reilly Media, Inc.", Sebastopol, California.

[63] Gabriel Skantze. 2003. Exploring human error handling strategies: Implications for spoken dialogue systems.

[64] Adam J Sporka, Sri H Kurniawan, Murni Mahmud, and Pavel Slavík. 2006. Non-speech input and speech recognition for real-time control of computer games. In *Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility*. ACM, New York, NY, USA, 213–220.

[65] Bernhard Suhm, Brad Myers, and Alex Waibel. 2001. Multimodal error correction for speech user interfaces. *ACM transactions on computer-human interaction (TOCHI)* 8, 1 (2001), 60–98.

[66] Marc Swerts, Diane Litman, and Julia Hirschberg. 2000. Corrections in spoken dialogue systems.

[67] Turn 10 Studios. 2013. *Forza Motorsport 5.* Game [XBox One]. Microsoft Studios, Redmond, Washington, U.S.

[68] Markku Turunen, Jaakko Hakulinen, K-J Raiha, E-P Salonen, Anssi Kainulainen, and Perttu Prusi. 2005. An architecture and applications for speech-based accessibility systems. *IBM Systems Journal* 44, 3 (2005), 485–504.

[69] Keith Vertanen and Per Ola Kristensson. 2010. Getting it right the second time: Recognition of spoken corrections. In *2010 IEEE Spoken Language Technology Workshop*. IEEE, IEEE, New York, NY, USA, 289–294.

[70] Marco Filipe Ganança Vieira, Hao Fu, Chong Hu, Nayoung Kim, and Sudhanshu Aggarwal. 2014. PowerFall: a voice-controlled collaborative game. In *Proceedings of the first ACM SIGCHI annual symposium on Computer-human interaction in play*. ACM, New York, NY, USA, 395–398.

[71] Tom Wilcox, Mike Evans, Chris Pearce, Nick Pollard, and Veronica Sundstedt. 2008. Gaze and voice based game interaction: the revenge of the killer penguins. *SIGGRAPH Posters* 81, 10.1145 (2008), 1400885–1400972.

[72] Chauncey Wilson. 2013. Interview techniques for UX practitioners: A user-centered design method.

[73] Rainer Winkler, Matthias Söllner, Maya Lisa Neuweiler, Flavia Conti Rossini, and Jan Marco Leimeister. 2019. Alexa, Can You Help Us Solve This Problem?: How Conversations With Smart Personal Assistant Tutors Increase Task Group Outcomes. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI EA '19)*. ACM, New York, NY, USA, Article LBW2311, 6 pages. https://doi.org/10.1145/3290607.3313090

[74] Beste F. Yuksel, Penny Collisson, and Mary Czerwinski. 2017. Brains or Beauty: How to Engender Trust in User-Agent Interactions. *ACM Trans. Internet Technol.* 17, 1, Article 2 (Jan. 2017), 20 pages. https://doi.org/10.1145/2998572

[75] Nima Zargham, Michael Bonfert, Georg Volkmar, Robert Porzel, and Rainer Malaka. 2020. Smells Like Team Spirit: Investigating the Player Experience with Multiple Interlocutors in a VR Game. In *Extended Abstracts of the 2020 Annual Symposium on Computer-Human Interaction in Play (CHI PLAY EA '20)*. ACM, New York, NY, USA, 408–412.

[76] Rui Zhao, Kang Wang, Rahul Divekar, Robert Rouhani, Hui Su, and Qiang Ji. 2018. An immersive system with multi-modal human-computer interaction. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*. IEEE, IEEE, New York, NY, USA, 517–524.