Design Patterns for Voice Interaction in Games

Fraser Allison[1], **Marcus Carter**[2], **Martin Gibbs**[1], **Wally Smith**[1]
[1]Interaction Design Lab, School of Computing and Information Systems, The University of Melbourne, Parkville, VIC, Australia
[2]Department of Media and Communications, The University of Sydney, Camperdown, NSW, Australia
[1]{fraser.allison,martin.gibbs,wsmith}@unimelb.edu.au, [2]marcus.carter@sydney.edu.au

ABSTRACT
Voice interaction is increasingly common in digital games, but it remains a notoriously difficult modality to design a satisfying experience for. This is partly due to limitations of speech recognition technology, and partly due to the inherent awkwardness we feel when performing some voice actions. We present a pattern language for voice interaction elements in games, to help game makers explore and describe common approaches to this design challenge. We define 25 design patterns, based on a survey of 449 videogames and 22 audiogames that use the player's voice as an input to affect the game state. The patterns express how games frame and structure voice input, and how voice input is used for selection, navigation, control and performance actions. Finally, we argue that academic research has been overly concentrated on a single one of these design patterns, due to an instrumental research focus and a lack of interest in the fictive dimension of videogames.

CCS CONCEPTS
• Applied computing~Computer games
• Computing methodologies~Speech recognition

Author Keywords
Design patterns; pattern language; interaction design; game design; voice interaction; voice control; speech recognition.

INTRODUCTION
Many visions of future technology, and future videogames, have us talking to virtual characters. In the film *Her* (2013), the main character falls in love with a Siri-like talking virtual assistant and spends his downtime playing a videogame with an alien character who communicates through foul-mouthed natural language conversation. In *2001: A Space Odyssey* (1968), an astronaut plays voice-command chess against his spaceship's talking computer. These scenarios are characterised by the fluency and ease of communication between humans and computer agents, which supports engaging and enjoyable gameplay. But this remains a hallmark of fiction, as the current reality of voice interaction gameplay is not nearly as gratifying. Feedback on voice interaction games often criticises the experience as feeling "unnatural" or "forced" [13:268], highlighting that game designers have found it challenging to design a satisfying player experience of voice interaction.

To help designers work in this space, and to crystallise some of the common approaches that have been taken in the past, we propose a pattern language for voice interaction game design. A pattern language is a collection of design patterns, each one of which describes the core of a common solution to a recurring design problem [1]. It is a practical vocabulary for classifying a design space, which formalises a body of knowledge that is otherwise implicit in the commonalities between finished works. Originating in the field of architecture, design patterns gained widespread popularity in computer science as a way of sharing reliable, repeatable approaches to common design goals [31]. Björk et al. [9] introduced pattern languages to game studies as a tool to help understand the form of games and to identify design choices that can affect the player experience in predictable ways.

In this paper, we present a pattern language for the use of voice interaction in games, based on a comprehensive survey of academic and commercial games that have used any form of voice interaction from 1973 to 2018. It catalogues the major game mechanics, dialogue structures and diegetic framing devices that have been employed, to give designers and scholars a library of options for addressing the challenges of designing in this notoriously difficult modality. We identify clusters of patterns that are often used together, and patterns that have so far resisted being used in combination with any other. We find that human-computer interaction (HCI) research to date has focused on only a narrow selection of design patterns, concentrated around pronunciation exercises, and paid little attention to the fictive and experiential elements that provide players with a sense of presence in a gameworld.

RELATED WORK

Voice Interaction in Digital Games
Research on voice control of digital games has been undertaken since at least the 1970s, more than a decade before voice control first appeared in commercial videogames [2]. Academic interest in the topic has predominantly come in the form of instrumental research

ACM 978-1-4503-5624-4/18/10…$15.00
https://doi.org/10.1145/3242671.3242712

[14], using game design to create an engaging format for language learning practice [39,49] and speech rehabilitation exercises [54,69]. Alongside this, a substantial body of research has explored the implementation of voice input as an alternative game control scheme, to enable access for players with motor impairments or other disabilities that prevent them from using physical controls [33,68]. Several research prototypes have explored options for controlling games through non-speech qualities of voice, such as the volume [76] and pitch [36,59,68] of vocal input; these approaches lend themselves to relatively simple arcade-style game mechanics such as one-dimensional movement.

Three-dimensional gameworlds introduce a particular challenge for voice interaction. Wadley and Ducheneaut [79] found that players struggle to communicate spatial deixis (referential concepts such as "there" and "on your left") when speaking to other human players in a virtual world. This difficulty is exacerbated with AI characters, which require a situation model of the game space before they can interpret commonplace spatially deictic phrases such as "come over here and pick this up" [32:140]. Several of the patterns that we identify (notably *Waypoints*, *Absolute Directions*, *Relative Directions*, *Select by Name* and *Select by Type*) are aimed at resolving such referential ambiguity without requiring complex situation models.

Studies on the player experience of voice interaction games have tended to emphasise its social character. Fletcher and Light describe the karaoke game as a "glue technology that assists in crafting and strengthening social linkages amongst players" [29:1]. Conversely, Dow et al. highlight the "uncomfortable" and "awkward" [26:9] feeling that some players experienced when talking to a computer. A study by Carter et al. found that players' feelings of embarrassment and awkwardness were most pronounced in games that required them to say things that were inconsistent with their in-game persona as the player-character – an effect they termed "identity dissonance" [13]. All together, these studies suggest that social framing is an important element in voice interactions in games. Consequently, we have captured the social framing of voice actions in our analysis of game design patterns.

In a previous study [2], we traced the history of voice interaction games from the 1970s to 2016, and noted characteristic differences between Japanese and Western voice interaction games. We identified the beginnings of a boom in small-scale games that focus on voice interaction as their central modality, and predicted that this boom would continue thanks to the increasing availability of microphone-equipped game platforms and effective speech recognition systems. The current study shows that this has eventuated, with 106 new voice interaction videogames released in 2017 alone, more than twice as many as any previous year. That excludes audio-only games, such as those on Amazon Alexa, of which there have been hundreds released since 2015. However, most of these recent games employ only a small set of simple and repetitive game mechanics. This context makes it even more timely to consider a pattern language that can illuminate the wider design space for voice interaction gameplay.

Design Patterns

Design patterns are a way of identifying and describing common solutions to design problems. The term was first used by Christopher Alexander in relation to architecture and urban planning [1], and has since been applied to interaction design fields such as human-robot interaction [41] and game design [8]. A design pattern consists of a formalised explanation of a repeatable approach to a specific design scenario. The elements of a pattern vary from Alexander's original formulation, but generally include an idiomatic title, an explanation of the problem or goal addressed, a description of the solution, one or more illustrative examples, and a note on connections with other patterns. Importantly, design patterns are not meant to be detailed blueprints, and do not include specific rules for how they should be implemented. Rather, Alexander conceptualises patterns as abstractions of a design solution:

> Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use the solution a million times over, without ever doing it the same way twice. [1:x]

Design patterns were introduced to the games research literature by Björk et al. [9]. They avoided framing patterns as solutions to problems, as this "creates a risk of viewing patterns as a method for only removing unwanted effects of a design" [9:185], rather than as a tool for more creative design work. Instead, they identify their value as tools for inspiring game ideas, understanding how existing games are put together, and helping to make choices about how new game will work and what elements they should include [8]. Of course, patterns like *Power Ups* and *High Score Lists* have been known in game design for decades, but collecting and formalising these patterns creates a sort of reference library in which the concepts can be detailed, annotated and cross-referenced for the benefit of designers who may be less familiar with their use. It also establishes a common language for discussing voice interaction, with clear points of reference to ensure mutual understanding [28].

Björk et al. [9] define a five-part format for game design patterns, comprising a short, specific and idiomatic *name*; a concise *description*, with examples; a discussion of *consequences* of applying the pattern; common choices involved in *using the pattern*; and a list of *relations* between patterns. Variations on this format have been used to collect patterns in game dialogue [11], character design [44,45,50,62], level design [35,52,67] and psychological factors in game design [24,46,51,80]. Pattern languages have often been used to map relatively under-documented design spaces, including mobile games [22], pervasive games [10], AI-based games [17,71], public games [7], disaster response games [70] and games controlled by the player's gaze [75].

Most pertinent for the current study is Alves and Roque's pattern language for sound design in games [3,4], which covers a wide variety of game sound patterns such as overheard conversations, contextual music and sound effects that signal player failure or success. While their collection is primarily concerned with sound output, it includes one pattern (*Sound Input*) that covers player-produced sound, including both sounds introduced to the game through a microphone and sounds made by the game in response to player button presses. This can be seen as an umbrella category for our study, which only concerns design choices related to sounds made by the player, and considers sound output only insofar as it relates to the uses of sound input.

Despite the enthusiasm for pattern languages, some cautions have been sounded about how they are applied. Erickson warns against the "common misconception" [28:361] that a pattern language is a prescriptive tool that describes universal templates for designers to apply directly from the page. The prescriptive approach is articulated by Dormans, who argues that "a design pattern library should depart from a clearly formulated notion of quality in games" [25:4] and present its patterns as prescribed solutions to common problems, on the pragmatic basis that this would be more attractive to game developers than a pattern language that does not pre-judge the quality of a design choice. Erickson counters this view with the argument that the purpose of a pattern language is generative: it provides a common meta-language to bootstrap the creation of a project-specific vocabulary in which all participants in a design process can effectively participate. Rather than being rigid prescriptions, "both the language and individual patterns are malleable and are used to generate site-specific pattern languages for particular projects" [28:361]. This is echoed in an interview study with HCI researchers [58], which found that overly formalised or prescriptive pattern languages tend to be less usable as they fail to account for the dynamic and contextual nature of design and technology. The study identified the extensive effort involved in collecting patterns and developing them into a coherent language as a barrier to their wider use, but once formulated, a pattern language was considered to be a particularly useful tool for capturing interactions, documenting conventions, sharing knowledge and encouraging design thinking.

A case study by Segerståhl and Jokela [65] noted shortcomings in the comprehensiveness, naming and organisation of some existing pattern languages which reduced their usability to designers. To improve the organisation of pattern languages, they advise that patterns should be grouped into task-related areas, such as **Searching** and **Navigation**, and that pattern names should be illustrative of the solution they represent (as in *Bread Crumbs*) rather than generic or ambiguous (as in *Main Navigation*). They emphasise the importance of examples to provide an illustration of the pattern in practice. Finally, they recommend integrating pattern collections into larger libraries with a consistent format. We have followed these suggestions as much as possible in the formation of the pattern language in this study.

NOTE ON TERMINOLOGY

Discussions about voice interaction and game characters share a common problem: their key terms are often used interchangeably, as they appear superficially similar, despite referring to distinct concepts [5,15]. To avoid ambiguity, in this section we define how we use these terms in this paper.

Voice interaction encompasses any way that voice can be used as an input modality for a computer system to process and render a response. Notably, it excludes player-to-player communication, as in online voice chat. *Voice control* refers to the intentional use of voice to directly control a system, which may be verbal or non-verbal, whereas *voice command* is specifically the use of spoken words as direct instructions to the system. A drone that moves in response to whistling would therefore be an example of voice control, but not voice command. Voice command relies on *speech recognition*, the recognition of spoken words by a computer system.

Virtual character encompasses any kind of character with an identity in a gameworld, whether controlled by a player or by AI, and whether or not it appears on screen. *Player character* is the specific virtual character that represents the player's identity in the gameworld, often with its own separately defined identity in the game fiction. It may appear in the form of an *avatar*, a virtual body over which the player exerts consistent control in the gameworld and serves as their locus of manipulation, "a tool that extends the player's ability to realise affordances within the game-world" [5:2]. All game characters that are not representative of or controlled by the player are referred to as *non-player characters* (NPCs); these are usually proximally controlled by an AI system rather than the player's inputs. Finally, *unit* is a generic term for AI-driven entities that includes those that are not strictly "characters", such as a group of NPCs that functions as one unified company, or a tank that is interchangeable with all other tanks. The term is associated with strategy games, in which characters often operate more like tokens on a game board than fully-realised characters.

Diegetic elements are part of the fiction of the gameworld; they are "real" from the perspective of the characters in the game. *Non-diegetic* elements are external to the gameworld, such as game menus or loading screens.

METHOD FOR COLLECTING DESIGN PATTERNS

To begin our design pattern collection, we first set out to identify every screen-based digital game that has used voice interaction up to early 2018. We compiled this list by conducting keyword searches on terms related to voice interaction and speech recognition in the online game libraries Steam, Giant Bomb, itch.io, Newgrounds, Kongregate and the Video Game Console Library, the review aggregator site Metacritic, the discussion website Reddit, and general web searches through Google Search. Next, we applied our search terms to Google Scholar and the ACM Digital Library to identify academic works that have described a unique voice interaction game or game control system. We looked at the prior works cited in these papers and, through their Google Scholar listings, works that had cited them since their publication to find more examples. We repeated this process with new keywords and venues, informed by our initial results, until our list converged on a set of 40 academic research games and 409 commercial and independent games published between 1973 and 2018.[1] English was by far the most common language these games used, but a substantial number were alternatively or exclusively available in other languages, primarily Japanese and a variety of European languages. Where our searches turned up non-English sources that described these games, we reviewed them using Google Translate. However, this study relies in the main on English-language sources.

For each game in our list, the first author recorded the origin, platform and game genre, how central voice interaction was to the gameplay, whether verbal or non-verbal voice input was recognised, what types of speech acts were possible, whether voice actions were diegetically framed as coming from the player or their avatar, and whether voice actions were directed towards the game system or objects within the gameworld. The last two categories were prompted by studies of voice interaction [13] and voice communication [78] that have found that voice can disrupt players' sense of presence and identity when it conflicts with the diegetic framing of the gameworld. The first author then provided a descriptive summary of each game's voice interaction mechanics, and identified potential patterns through an open-coding process. Initial codes described the specific combination of a voice input and a response in the game (such as *Blow air to move sailing ship*). As shared themes emerged across the data set, the codes were iteratively revised and abstracted to describe higher-level patterns (such as *Breath physics*). We applied additional codes to describe how voice actions were framed by the game's presentation (see **Patterns for Diegetic Framing**). We ceased coding once we arrived at a stable list of patterns that encapsulated all the main variations we observed in our list of games.

After this process was complete, we conducted a smaller survey on 22 audio-only games on Amazon Alexa, to test whether different design patterns were apparent. The patterns we found were consistent with those in the larger survey. Our findings for audiogames are discussed at the end of the paper.

[1] A complete list of the games surveyed for this study has been submitted as a separate spreadsheet.

VOICE INTERACTION GAME DESIGN PATTERNS

Our pattern language comprises 25 patterns for voice interaction (see table 1). These patterns are descriptions of emergent features of this design landscape. They are not intended to be exhaustive or mutually exclusive; this is a language, not a logical model for the categorisation of voice interaction games. Like a language, it resists neat categorisations and ontological consistency. However, to aid understanding we have arranged the pattern into six categories, which we will now explain.

1 Diegetic framing	2 Dialogue structure	3 Selection
Speak as a character	Choice of options	Select by name
Speak as the player	Question and answer	Select by type
Situated commander	Who-what-where commands	
Floating commander	Scripted conversation	
	Unscripted conversation	

4 Navigation	5 Control	6 Performance
Waypoints	Name an action	Pronunciation
Absolute directions	Name a behaviour	Karaoke
Relative directions	Shout to shoot	Breath physics
	Volume control	Exclamation
	Pitch control	Spellcasting
		Overheard noise

Table 1. Game design patterns for voice interaction.

The first category, **Diegetic Framing**, consists of the patterns for how the game fiction frames the player's voice input. The patterns in this category are relatively abstract, as they describe the collective implications of multiple game elements, from narrative text to game mechanics. They address the role-playing dimension that has been identified as an important factor in players' experience of voice interaction [2,13].

The second category, **Dialogue Structure**, contains the patterns that describe how dialogue is configured between a player and an NPC. This category looks one level above the individual utterances, to the arrangement of utterances into a particular form of dialogue, such as conversation.

The remaining patterns are categorised into four broad task areas, in accordance with Segerståhl and Jokela's [65] guidelines for usable pattern languages. The categories are **Selection**, **Navigation**, **Control** and **Performance**. Patterns therein describe individual utterances and the responses they engender from the game.

We define each pattern with an illustrative *Name*, a short *Description*, an *Example* that exemplifies the pattern, a brief discussion of *Consequences* for the player experience, and a compact list of *Relations* to other patterns, such as conflicts and parent-child relationships. This is a simplified version of Björk et al.'s [9] model, foregoing their section on *Using the Pattern* for brevity, to allow for a more comprehensive list of the design patterns we have observed.

We distinguish patterns by how they are presented to the player, rather than by their underlying mechanism in the game system. To illustrate this, consider a game that asks the player to extinguish a candle by blowing into the microphone in one scene, and to interrupt a wedding by shouting an objection in another scene. The trigger for both actions may be identical in the game code, with either one being activated by any sufficiently loud sound. But in our pattern language, the former would be described as *Breath Physics* and the latter as *Exclamation* in accordance with their diegetic framing.

1. Patterns for Diegetic Framing

Patterns in this category describe how the game fiction frames and positions the player's voice actions in relation to the game's imaginary. These patterns do not describe specific voice actions, but how the configuration of voice actions casts the player in a speaking role. The patterns are *Speak as a Character*, *Speak as the Player*, *Situated Commander* and *Floating Commander*.

1.1 Speak as a Character

The player's utterances correspond to the utterances of a character in the gameworld. The game responds to the player's voice as though it is the diegetic voice of the player-character.

Examples: *Guitar Hero: World Tour* [55] represents players as a rock band on stage, performing the song that the players are performing. The virtual crowd cheers or heckles the virtual singer based on how well the player sings.

Consequences: Maintains consistency between the player's in-game persona and the persona implied by their voice actions. Player commentary suggests that this supports an experience of immersion and flow [13].

Relations: Instantiated by *Situated Commander* and *Floating Commander*. Instantiated by patterns that represent embodied diegetic actions, namely *Scripted Conversation*, *Unscripted Conversation*, *Spellcasting*, *Breath Physics*, *Overheard Noise* and often *Exclamation*. Conflicts with *Speak as the Player*.

1.2 Speak as the Player

The player's utterances do not correspond to the utterances of a character in the gameworld. The game treats the player's voice actions as non-diegetic inputs.

Example: In *Tomb Raider: Definitive Edition* [20], the player can say the name of a weapon to make the player-character hold that weapon. This would not make sense as a vocalisation by the player-character, as she would be giving verbal instructions to herself.

Consequences: Carter et al. [13] noted that players sometimes experienced "identity dissonance" when speaking to (rather than as) the player-character, which suggests that *Speak as the Player* can disrupt the player's sense of presence in the gameworld if not applied carefully.

Relations: Conflicts with *Speak as a Character* and patterns that instantiate it: *Situated Commander*, *Floating Commander*, *Scripted Conversation*, *Unscripted Conversation*, *Spellcasting*, *Breath Physics*, *Overheard Noise* and often *Exclamation*.

1.3 Situated Commander

The player gives voice commands to NPCs in the role of an on-screen avatar who they control directly.

Example: This pattern is common in tactical shooter games, such as *Rainbow Six: Vegas* [73], in which the player controls the leader of a military squad and can give the other squad members instructions such as "check your fire" and "regroup on me".

Consequences: Provides the experience of an embodied persona as the imagined source of the player's voice, but also requires the player to divide their attention between managing the avatar's actions and using voice. This can be challenging, especially when the avatar is under threat or time pressure, as speech formulation draws on some of the same cognitive resources that are needed for strategic problem solving, to a greater extent than hand-eye coordination does [66].

Relations: Instantiates *Speak as a Character*. Can be instantiated by *Who-What-Where Commands*. Conflicts with *Speak as the Player* and *Floating Commander*.

1.4 Floating Commander

The player gives voice commands to NPCs from a free-floating perspective above the gameworld, in the role of an unseen player-character.

Example: In *Tom Clancy's EndWar* [74], the player directs a battalion of infantry, tank and helicopter squads from above the battlefield using *Who-What-Where Commands*.

Consequences: Allows the player to dedicate their full attention to issuing commands without the distraction of managing an avatar. Combined with the wide field of view granted by the bird's-eye perspective, this makes *Floating Commander* well suited to strategy games and games with complex *Who-What-Where Commands*.

Relations: Instantiates *Speak as a Character*. Can be instantiated by *Who-What-Where Commands*. Conflicts with *Speak as the Player* and *Situated Commander*.

2. Patterns for Dialogue Structure

Patterns in this category describe a systematic arrangement of speech between the player and the game. They are concerned with how utterances are configured into a certain form of dialogue. The patterns are *Choice of Options*, *Question and Answer*, *Who-What-Where Commands*, *Scripted Conversation* and *Unscripted Conversation*.

2.1 Choice of Options

The game presents a list of options that the player can select from by saying an associated phrase. The phrase may be the full text of the option or a corresponding keyword.

Example: In *Thayer's Quest* [61], each scene ends with a numbered list of locations. The player selects a destination by saying the number next to the place they want to visit.

Consequences: Reduces the possible inputs, which allows the speech recognition task within the game system to be simplified to improve reliability. Also simplifies the task for the player, minimising uncertainty and the need for creative cognition to formulate an answer.

Relations: Instantiated by *Scripted Conversation* and often *Question and Answer*.

2.2 QUESTION AND ANSWER

A series of questions and responses between the player and a game character (or the game interface). Either party may be the questioner.

Example: In the racing game *Forza Horizon 2* [60], a voice-activated virtual assistant named ANNA provides verbal directions on request. If the player asks "ANNA, what should I do?", ANNA may say "You haven't attempted a Bucket List Challenge for a while, would you like me to direct you to the nearest one?" If the player responds "Yes", ANNA gives turn-by-turn directions to the event.

Consequences: Replicates the familiar dynamic of a virtual assistant such as Siri or Alexa. Provides a relatively clear and constrained task for the game system. When the game character acts as the questioner, their questions can steer the player into giving certain kinds of responses that the game can recognise.

Relations: Often instantiates *Choice of Options*. Can instantiate *Scripted Conversation* or *Unscripted Conversation*.

2.3 WHO-WHAT-WHERE COMMANDS

The game accepts voice commands in the format of an imperative phrase that directs a subject (who) to take an action (what) in a place (where), in that order. The "who" or "what" element is sometimes unstated when the target subject or place is implicit in the rest of the command phrase or due to the context.

Example: *There Came an Echo* [38] uses an extensive vocabulary of *Who-What-Where Commands*, such as "Miranda move to Alpha Six" and "Everybody focus fire on Target Two on my mark". As these examples show, the commands can incorporate *Select by Name* or *Select by Type* to denote their subject or object, *Name an Action* or *Name a Behaviour* to denote their action, and *Waypoints* to denote their location. The second example also shows how *There Came an Echo* adds a "when" element to the pattern: the command is inactive until the player next says "mark".

Consequences: Allows the player to convert a small vocabulary of keywords into an extensive and highly configurable set of instructions, in a format that mimics everyday speech. This makes it well suited to games that focus on coordinating the movement and actions of multiple characters around a space, notably strategy games. This pattern creates the narrative implication that the player-character is in a position of authority over other characters, and seeing characters obey verbal orders supports a player experience of authority as well.

Relations: Can instantiate *Situated Commander* or *Floating Commander*. Can be instantiated by some combination of *Select by Name*, *Select by Type*, *Name an Action*, *Name a Behaviour*, *Waypoints*, *Relative Directions* and *Absolute Directions.*

2.4 SCRIPTED CONVERSATION

An in-character conversation in which the player participates by reading pre-scripted lines of dialogue that appear on screen, effectively "performing" the voice of the player-character.

Example: During conversation scenes in *Mass Effect 3* [6], each time the player-character has a turn to talk, the game shows a list of several options for what they can say. The player can choose an utterance from this list by reading it aloud. The player-character then gives a longer version of this utterance in its own voice, and the NPC responds.

Consequences: Allows the player to participate in a fully expressive conversation without the need for a sophisticated language understanding system, as the game system only needs to match their speech to a small number of pre-set utterances. Pre-scripted responses remove the task of formulating a response in a phrase the game will accept, which can interfere with a player's sense of presence [64].

Relations: Instantiates *Speak as a Character* and *Choice of Options*. Can be instantiated by *Question and Answer*. Conflicts with *Unscripted Conversation*.

2.5 UNSCRIPTED CONVERSATION

An in-character conversation in which the player participates by formulating and speaking utterances in the role of the player-character.

Example: Gameplay in *Seaman* [77] centres on *Unscripted Conversation* with a talking fish. The fish asks questions like "What do you do for a living?" and the player is given no specifications for how to answer. The game system listens for relevant keywords to match the player's answer with an appropriate response from the fish.

Consequences: This is the voice interaction format that most closely approximates everyday speech, and the one that is often imagined in depictions of future technology. Formulating an appropriate natural-language utterance on the spot can be more enjoyable than choosing from a menu of options, but it is also a more difficult task, and can leave the player with a lower sense of control or involvement in the story [64].

Relations: Instantiates *Speak as a Character*. Can be instantiated by *Question and Answer*. Conflicts with *Unscripted Conversation*.

3. Patterns for Selection

Patterns in this category describe methods of distinguishing between objects, including units and characters, with voice input. The patterns are *Select by Name* and *Select by Type*.

3.1 SELECT BY NAME

The player selects a specific object in the gameworld by saying its unique name.

Example: In *There Came an Echo*, the voice command "Miranda attack Target Two" uses *Select by Name* to identify both its subject (the friendly character Miranda) and its object (the enemy character labelled Target Two).

Consequences: One of the major limitations of voice interaction is its lack of a precise locus of manipulation, like a mouse pointer, which makes it difficult to quickly and precisely select an object on screen [79]. Unique names overcome this by providing an unambiguous and stable link between a game object and a specific utterance.
Relations: Instantiates *Who-What-Where Commands* in combination with *Absolute Directions*, *Relative Directions*, *Name an Action* or *Name a Behaviour*. Instantiated by *Waypoints*.

3.2 SELECT BY TYPE
The player selects objects of a certain type by saying a category to which they belong. May allow multiple categories to narrow down the selection.
Example: *Tom Clancy's EndWar* uses this pattern to select every unit in a category, with commands such as "calling all tanks, retreat". *I'm Hiding* [40] uses it to identify individual objects, through a combination of three words that collectively describe only one object on the screen, such as "big green crayon".
Consequences: Like *Select by Name*, this pattern helps to connect player speech to objects in the gameworld, with the added flexibility of enabling reference to multiple objects at once, and without the need to give every selectable object a unique name. However, it is more ambiguous than *Select by Name* and requires more thinking, as the player needs to appraise the qualities of the object and match them to a valid category label; for example, "green crayon" rather than "emerald marker".
Relations: Instantiates *Who-What-Where Commands* in combination with *Absolute Directions*, *Relative Directions*, *Name an Action* or *Name a Behaviour*.

4. Patterns for Navigation
Patterns in this category describe methods for delineating spaces and directions through voice input. The patterns are *Waypoints*, *Absolute Directions* and *Relative Directions*.

4.1 WAYPOINTS
The player can select a specific location in the gameworld by saying its unique name or set of coordinates.
Example: *ATC Voice* [47] uses the names of real-world air traffic control radio beacons as waypoints. The command "Learjet eight two golf fly direct blue mesa VOR" would tell a Learjet plane with the identification number 82G to fly towards a radio beacon named Blue Mesa.
Consequences: Allows the player to quickly and precisely indicate a location through voice input. Also reduces the decision space by having a limited set of locations that afford selection. This constricts the player's options and simplifies their decision-making process for navigation, which could be useful or frustrating depending on the task.
Relations: Instantiates *Select by Name*. Instantiates *Who-What-Where Commands* in combination with *Select by Name*, *Select by Type*, *Name an Action* or *Name a Behaviour*.

4.2 ABSOLUTE DIRECTIONS
The player can command a unit to move in a direction relative to the orientation of the gameworld.

Example: *ATC Voice* uses compass headings and altitudes to direct planes in flight. The command "Turn left heading zero nine zero, climb maintain one one thousand" tells a plane to turn directly East (90 degrees from magnetic North) and ascend to an altitude of 11000 feet.
Consequences: One of the rarest patterns for voice interaction. We suggest this is because it conflicts with players' tendency to focus on the orientation of their character, making *Relative Directions* more salient to the player's perspective in many cases.
Relations: Instantiates *Who-What-Where Commands* in combination with *Select by Name*, *Select by Type*, *Name an Action* or *Name a Behaviour*.

4.3 RELATIVE DIRECTIONS
The player can command a unit to move in a direction relative to the unit's own orientation.
Example: In *Northern Lights* [21], the player controls a dog sled driver, shouting "left", "right", "forward" and "stop" to tell the sled dogs where to go.
Consequences: Approximately replaces the role of the analogue stick or directional buttons in traditional game controls, but without the capacity for continuous variable input that these controls and *Volume Control* share.
Relations: Instantiates *Who-What-Where Commands* in combination with *Select by Name*, *Select by Type*, *Name an Action* or *Name a Behaviour*.

5. Patterns for Control
Patterns in this category describe ways of using voice input to control the actions of game characters. Patterns in this category imply some separation between the actions of the player and the actions of the characters in the game. The patterns are *Name an Action*, *Name a Behaviour*, *Shout to Shoot*, *Volume Control* and *Pitch Control*.

5.1 NAME AN ACTION
The player says the name of a discrete action to make a character perform that action.
Example: In *Nintendogs* [56], the player can tell a puppy to "lie down" or "roll over" on command.
Consequences: The simplicity and directness of mapping the verb for an action to the action makes it an intuitive combination. Indeed, some game designers use the term "verbs" to describe player actions even when not referring to spoken actions [18]. Its popularity bears out this impression, as it is one of the most common patterns in our dataset, applied to most scenarios for voice interaction gameplay.
Relations: Instantiates *Who-What-Where Commands* in combination with *Select by Name*, *Select by Type*, *Waypoints*, *Absolute Directions* or *Relative Directions*. Can be instantiated by *Spellcasting*.

5.2 Name a Behaviour

The player says the name of a mode of behaviour to make a character adopt that behaviour.

Example: In the soccer game *FIFA 14* [27], the player can call out tactics such as "ultra-defensive" and "slow it down" to make their team shift behavioural priorities.

Consequences: Expands the voice command repertoire to encompass macro-instructions. In so doing, it distances the player's input from the game action and creates a more mediated sense of control, by placing a greater degree of decision-making for individual actions on the AI systems that control NPCs.

Relations: Instantiates *Who-What-Where Commands* in combination with *Select by Name* or *Select by Type*.

5.3 Shout to Shoot

When the player shouts, their avatar shoots.

Example: In the sidescrolling shoot-em-up *Pah!* [43], the player controls a spaceship flying horizontally at a constant speed. The ship fires a laser each time the player shouts "pah!" (or any word) above a certain volume threshold. Vocalisations below that threshold move the ship up and down on the screen, using the *Volume Control* pattern.

Consequences: Like *Volume Control*, this pattern is technically and conceptually simple but physically fatiguing for the player, especially if rapid shooting is required. It has almost exclusively been used in simple shoot-em-up games on mobile platforms, reflecting both its limited depth and its ability to create an amusing spectacle for an audience.

Relations: Usually conflicts with *Speak as a Character* and instantiates *Speak as the Player*, due to the dissociation between the acts of shouting and shooting.

5.4 Volume Control

The volume of the player's voice modulates the speed or direction of a unit's movement.

Example: In the side-scrolling platforming game *Yasuhati* [30], the avatar walks forward at a speed relative to the volume of the player's voice: the louder the sound, the faster the movement. When the volume rises above a certain level, the avatar jumps. The player must alternate between shouts to jump between platforms and quiet speech to walk across platforms.

Consequences: In a technical sense, this pattern uses the simplest form of voice interaction: converting the loudness of sound into a one-dimensional continuous variable input. For players, however, it is a difficult task to maintain precise control over their voice volume. And paradoxically, the fact that this pattern disregards speech content creates more difficulty, as it forces the player to make continual arbitrary decisions about what sounds to make. As a result, players often end up uttering nonsensical combinations of words, or meaningless gibberish. This combination of amusingly absurd speech and difficult gameplay has made *Volume Control* popular for playing in front of an audience, including on YouTube [53].

Relations: Tends to conflict with *Speak as a Character* and therefore instantiate *Speak as the Player*. Although it could in theory be combined with a speech recognition-based pattern, there are no examples of this in our dataset.

5.5 Pitch Control

The frequency of the player's voice modulates the speed or direction of a unit's movement.

Example: In Sporka et al.'s [68] humming-controlled Tetris variant, a hum with a rising pitch moves the puzzle piece right, a falling pitch moves the piece left, and a constant pitch moves the puzzle piece down.

Example: *Plurality Spring* [72] combines *Pitch Control* with *Match Pitch* and *Waypoints*. The screen displays several nodes that are each labelled with a series of musical notes. Matching the note sequences moves the avatar to the node.

Consequences: Affords a one-dimensional continuous variable input, like *Volume Control*, but with greater flexibility. Players can use quieter voice inputs to minimise fatigue and social embarrassment. Supports a "vocabulary" of controls based on pitch shifts and intervals, as in the examples above.

Relations: None.

6. Patterns for Performance

Patterns in this category describe voice actions that directly perform the activity they are meant to produce. In these patterns, unlike the previous category, the player enacts the in-game action with their own voice. The patterns are *Pronunciation*, *Karaoke*, *Breath Physics*, *Exclamation*, *Spellcasting* and *Overheard Noise*.

6.1 Pronunciation

The player must match the pronunciation of a word as closely as possible to succeed.

Example: Most common in serious games designed for language learning, such as *Word War* [49], or speech rehabilitation, such as *Talking to Teo* [54]. These games replicate classroom speech exercises to allow players to practice independently and get feedback on their efforts.

Consequences: Focuses the game on the act of speech recognition for its own sake, rather than as a means to a broader end.

Relations: None. While intelligible pronunciation is always needed for speech recognition to work, the *Pronunciation* pattern describes cases where it is the primary goal.

6.2 Karaoke

The player sings along to a song and is scored on the accuracy of their pitch. The game plays the backing music for the song and shows the lyrics on-screen, next to a visual indicator of their pitch.

Example: *Karaoke Revolution Volume 2* [34] was the first *Karaoke* game to include all the main elements that are now commonly found in this pattern, including a piano-roll style pitch indicator, handheld microphone, and virtual crowds on screen that respond to how well the player is singing.

Consequences: A notably social gameplay pattern, often played in front of an audience. Fletcher and Light's description of *Karaoke* as a "glue technology that assists in

crafting and strengthening social linkages amongst players" [29:1] highlights its focus on extroverted performance and the opportunities for self-expression that it facilitates.
Relations: None.

6.3 Breath Physics
The player blows into the microphone and the gameworld responds as though it felt the gust of wind.
Example: *The Legend of Zelda: Phantom Hourglass* [57] requires the player to blow into the microphone to extinguish small flames, as though blowing out a candle.
Example: *Resonance: The Lost Score* [23] uses a variation on this pattern, in which the player's voice can smash glass when it is held at the right frequency. In this case, it is the resonance rather than the sheer wind-force of the player's breath that physically affects the gameworld.
Consequences: An easy pattern for both the player and the game developer, as blowing air takes much less effort than shouting, but registers as a loud noise due to the sustained physical displacement of the microphone diaphragm. Even small and low-quality microphones can reliably detect air blowing into them, which makes this pattern technically viable on portable game devices.
As *Breath Physics* creates the illusion of directly physically affecting the gameworld, it should increase the player's feeling of presence in the gameworld. It is also likely to support the player's sense of identification with the player-character, as Carter et al. [13] describe, by aligning the actions of the player and the player-character.
Relations: Instantiates *Speak as a Character* and conflicts with *Speak as the Player*, by conflating the player with a diegetic player-character, even if the latter is only an implied presence in the gameworld.

6.4 Exclamation
The player exclaims loudly to trigger a vocal outburst by the player character.
Example: The adventure game *Phoenix Wright: Ace Attorney* [12] includes scenes of witness testimony in a courtroom. If the player yells "hold it!", "objection!" or "take that!" during these scenes, the player-character will interrupt the witness with the same shouted phrase, and then begins a cross-examination of what the witness just said.
Consequences: Allows the player to physically enact a moment of heightened emotion and drama, creating a strong (if momentary) opportunity for role-playing and identification with the character.
Relations: Instantiates *Speak as a Character* and conflicts with *Speak as the Player*, as the player's utterance is normally the same as the character's.

6.5 Spellcasting
The player performs a verbal incantation to invoke a magical effect in the gameworld. The incantation is often in the form of words from an invented language rather than a phrase in a real-world language.
Example: *In Verbis Virtus* [37] casts the player-character as a mage who casts spells from their hand by speaking phrases in the fictional Maha'ki language. Spells include "ekto lumeh" to project a beam of light and "obee kehnu" to lift and move an object telekinetically.
Consequences: The metaphor of *Spellcasting* is a useful conjunction between the tropes of the fantasy genre, which is common in games, and the effects of speech recognition technology. The pattern supports an embodied experience with a coherent sense of player identity while performing effects that would not normally be possible through speech.
Relations: Instantiates *Speak as a Character*. Conflicts with *Speak as the Player*. Can instantiate *Name an Action* if the incantation describes the effect of the spell.

6.6 Overheard Noise
When the player makes a sound, NPCs are alerted to the presence of the player-character.
Example: In the stealth horror game *Alien: Isolation* [19], the player-character must hide from an alien that is attracted towards in-game sounds, from loud gunshots to the quiet beeping of a motion tracker device. In the optional "noise detection" mode, the alien also responds to noise in the real world that is picked up by the game system's microphone, and reacts as though the sound had emanated from the player-character's position in the gameworld.
Consequences: Reduces the perception of a separation between the physical world and the gameworld [13]. This can increase the player's feeling of presence, but it can also disrupt it if sounds from other people or outside events are accidentally picked up by the game and interfere with the experience [78].
Relations: Instantiates *Speak as a Character*. Conflicts with *Speak as the Player*.

DISCUSSION
Through this pattern collection, we have provided an index of the main configurations that have been used for voice interaction in games. Its usefulness is primarily in defining a core vocabulary for describing voice interaction games, and in offering a list of tested solutions for designers who are approaching a voice-based game mechanic for the first time. However, it can also be used by scholars as a wider lens to survey the landscape of voice interaction games, and identify emergent conventions or neglected combinations.

When we do so, one of the first things that becomes apparent is that certain patterns have tended to cluster together. Most obvious is the "command" cluster: *Who-What-Where Commands* are always instantiated through some combination of *Select by Name*, *Select by Type*, *Waypoints*, *Absolute Directions*, *Relative Directions*, *Name an Action* and *Name a Behaviour*, and almost always framed by the persona of a *Situated Commander* or *Floating Commander*. This cluster is usually found in games that focus on tactically coordinating the movement of multiple units across a contested landscape, such as strategy games and team sports games. Most of the patterns in this cluster are solutions to a similar core problem: making an unbounded possibility space manageable through speech by assigning specific words as anchors for predefined action options.

A smaller cluster may be called the "branching dialogue" cluster, consisting of *Scripted Conversation*, *Choice of Options*, *Speak as a Character* and often *Question and Answer*. These patterns co-occur in games with dialogue scenes where the player (in the persona of the player-character) chooses what to say to an NPC, often asking questions that prompt some exposition about the game's narrative arc. This cluster is sometimes found in games that also feature the "command" cluster, as in *Mass Effect 3*.

Conversely, many patterns are notable for rarely or never appearing in combination with other patterns. *Shout to Shoot*, *Volume Control*, *Pitch Control*, *Pronunciation*, *Karaoke*, *Exclamation* and *Overheard Noise* are all notable for their seeming resistance to being used in combination with any other pattern. Something that distinguishes this set is that they are all patterns of direct action rather than mediated action: the player controls or performs a task directly with their voice, rather than using their voice to exchange information with a character, as in a conversation or a command. Whether these patterns' lack of integration with other patterns is a result of technical effort or because they serve different types of player experiences is not clear, but would be an interesting point for further study.

When we narrow our focus to the voice interaction patterns found in academic papers on games, we see a reflection of what Carter et al. call the operative paradigm of HCI games research [14]. That is to say, research has focused on using games instrumentally, as mechanisms for producing secondary benefits rather than for producing meanings or positive experiences directly. As such, games research has explored only a narrow set of voice interaction patterns. *Pronunciation* is particularly common among academic research games (e.g. [16,42,48,54,63,69]), as it provides a gamified format for language learning or speech therapy exercises. Accordingly, games in academic research are much more likely to use *Speak as the Player* than commercial and independent games; even those that include a central character are more likely to ask the player to talk *to* it than to talk *as* it. The patterns employed in these games shows that they are generally less engaged with the fictive and narrative elements of the videogame format, and do relatively little to give the player a sense of presence in a meaningful gameworld.

Audiogames

To create this pattern language, we sought to review the design of every screen-based digital game that has used voice interaction, and we are confident that our dataset includes almost all such games that are identifiable today. However, it excludes audiogames: digital games that have no significant visual component. This category has grown rapidly in the past three years with the expansion of the smart speaker market, such that there are now hundreds of apps for Google Home and thousands of skills for Amazon Alexa categorised as games. Audiogames merit a separate analysis, which our approach here is poorly suited to provide, due to the large number of minimally-documented audiogames, the fundamental difference of the format, and the fact that nearly all the academic literature to date has focused on voice interaction with screen-based games.

To test whether our pattern language is transferrable to audiogames, we surveyed 22 audiogames on Amazon Alexa that had been widely reviewed. *Choice of Options* was the core interaction pattern for 17 of these games, and 11 implemented their choices in a *Question and Answer* dialogue. Four games used *Who-What-Where Commands* as the core interaction pattern. Interestingly, four games used *Absolute Directions*, which was the rarest pattern in our main dataset. We interpret this as a sign that audiogames do not convey as strong a sense of orientation in the gameworld as screen-based games, due to the lack of a visible avatar for the player to orient themselves with.

CONCLUSION

In this paper, we have introduced the idea of a pattern language for voice interaction in games. We discussed the background of pattern languages in game design, reviewed studies on best practices for design patterns for HCI, and provided an overview of challenges that have been identified for voice interaction in games – primarily spatial ambiguity, and social awkwardness due to conflicts between the identity roles of the player and the player-character. We surveyed 449 videogames and 22 audiogames to discover 25 design patterns, related to diegetic framing, dialogue structure, selection, navigation, control and performance. We noted clusters of patterns around command and conversation, and a number of isolated patterns that used voice for direct action. Finally, we observed that academic research has focused on a narrow subset of design patterns, especially *Pronunciation*.

We believe the narrow focus of academic research partly reflects a discourse around voice interaction that is overly concerned with speech recognition, and with word recognition accuracy rates in particular, as a perceived barrier to a satisfying player experience. As our pattern language has demonstrated, this does not need to be the case. Game designers have been able to create enjoyable and popular game experiences using entirely non-verbal forms of voice interaction, with *Karaoke* and *Volume Control* being the most common examples to date but far from exhausting the possibilities of this modality. And patterns like *Spellcasting* and *Waypoints* have been used successfully to constrain speech input to clear, easily distinguishable phrases that allow imperfect speech recognition systems to recognise utterances with a high degree of accuracy. We hope that cataloguing these possibilities as design patterns encourages more experimentation with the format, and reminds both game designers and researchers of options that they have to take voice interaction in less obvious directions.

ACKNOWLEDGMENTS

Fraser Allison's work was supported by an Australian Government Research Training Program Scholarship and by the Microsoft Research Centre for Social NUI.

REFERENCES

1. Christopher Alexander, Sara Ishikawa, Murray Silverstein, Max Jacobson, Ingrid Fiksdahl-King, and Shlomo Angel. 1977. *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press, New York.
2. Fraser Allison, Marcus Carter, and Martin Gibbs. 2017. Word Play: A History of Voice Interaction in Digital Games. *Games and Culture*: 155541201774630.
3. Valter Alves and Licinio Roque. 2010. A Pattern Language for Sound Design in Games. In *Proceedings of the 5th Audio Mostly Conference: A Conference on Interaction with Sound* (AM '10), 12:1–12:8. https://doi.org/10.1145/1859799.1859811
4. Valter Alves and Licinio Roque. 2011. A Deck for Sound Design in Games: Enhancements Based on a Design Exercise. In *Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology* (ACE '11), 34:1–34:8. https://doi.org/10.1145/2071423.2071465
5. Peter Bayliss. 2007. Beings in the Game-world: Characters, Avatars, and Players. In *Proceedings of the 4th Australasian Conference on Interactive Entertainment* (IE '07), 4:1–4:6. Retrieved June 1, 2015 from http://dl.acm.org/citation.cfm?id=1367956.1367960
6. BioWare. 2012. *Mass Effect 3*. Electronic Arts, Redwood City, CA.
7. Staffan Björk. 2012. *Gameplay Design Patterns for Public Games*. EXPERIMEDIA.
8. Staffan Björk and Jussi Holopainen. 2004. *Patterns in Game Design*. Charles River Media, Inc., Rockland, MA.
9. Staffan Björk, Sus Lundgren, and Jussi Holopainen. 2003. Game Design Patterns. In *Proceedings of the 2003 DiGRA International Conference: Level Up* (DiGRA '03).
10. Staffan Björk and Johan Peitz. 2007. Understanding Pervasive Games through Gameplay Design Patterns. In *DiGRA '07 - Proceedings of the 2007 DiGRA International Conference: Situated Play*.
11. Jenny Brusk and Staffan Björk. 2009. Gameplay Design Patterns for Game Dialogues. In *Proceedings of DiGRA 2009*.
12. Capcom. 2005. *Phoenix Wright: Ace Attorney*. Capcom, Osaka, Japan.
13. 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* (CHI PLAY '15), 265–269. https://doi.org/10.1145/2793107.2793144
14. Marcus Carter, John Downs, Bjorn Nansen, Mitchell Harrop, and Martin Gibbs. 2014. Paradigms of Games Research in HCI: A Review of 10 Years of Research at CHI. In *Proceedings of the First ACM SIGCHI Annual Symposium on Computer-human Interaction in Play* (CHI PLAY '14), 27–36. https://doi.org/10.1145/2658537.2658708
15. Marcus Carter, Martin Gibbs, and Michael Arnold. 2012. Avatars, Characters, Players and Users: Multiple Identities at/in Play. In *Proceedings of the 24th Australian Computer-Human Interaction Conference* (OzCHI '12), 68–71. https://doi.org/10.1145/2414536.2414547
16. Gabriel J. Cler, Talia Mittelman, Maia N. Braden, Geralyn Harvey Woodnorth, and Cara E. Stepp. 2017. Video Game Rehabilitation of Velopharyngeal Dysfunction: A Case Series. *Journal of Speech, Language, and Hearing Research* 60, 6S: 1800–1809. https://doi.org/10.1044/2017_JSLHR-S-16-0231
17. Michael Cook, Mirjam Eladhari, Andy Nealen, Mike Treanor, Eddy Boxerman, Alex Jaffe, Paul Sottosanti, and Steve Swink. 2016. PCG-Based Game Design Patterns. *arXiv:1610.03138 [cs]*. Retrieved February 27, 2018 from http://arxiv.org/abs/1610.03138
18. Chris Crawford. 2004. *Chris Crawford on Interactive Storytelling*. New Riders, Berkeley, CA.
19. Creative Assembly. 2014. *Alien: Isolation*. Sega, Tokyo.
20. Crystal Dynamics. 2014. *Tomb Raider: Definitive Edition*. Square Enix, Tokyo.
21. Augustin Dagoret, Jean-Louis Boudrand, Lorris Giovagnoli, Nicolas Terlon, Aymeric Thevenot, and Marie Vilain. 2016. *Northern Lights*. École Nationale du Jeu et des Médias Interactifs Numériques, Angoulême, France.
22. Ola Davidsson, Johan Peitz, and Staffan Björk. 2004. *Game Design Patterns for Mobile Games*. Nokia Research Centre.
23. Demerara Games. 2017. *Resonance: The Lost Score*. Demerara Games, Natal, Brazil.
24. Claire Dormann, Jennifer R. Whitson, and Max Neuvians. 2013. Once More With Feeling: Game Design Patterns for Learning in the Affective Domain. *Games and Culture* 8, 4: 215–237. https://doi.org/10.1177/1555412013496892
25. Joris Dormans. 2013. Making Design Patterns Work. In *Proceedings of the Second Workshop on Design Patterns in Games, DPG*.
26. 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* (CHI '07), 1475–1484. https://doi.org/10.1145/1240624.1240847
27. EA Canada. 2013. *FIFA 14*. EA Sports, Redwood City, CA.
28. Thomas Erickson. 2000. Lingua Francas for Design: Sacred Places and Pattern Languages. In *Proceedings of the 3rd Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques* (DIS '00), 357–368. https://doi.org/10.1145/347642.347794
29. Gordon Fletcher and Ben Light. 2011. Interpreting digital gaming practices: SingStar as a technology of work. Retrieved March 30, 2015 from http://usir.salford.ac.uk/17245/
30. Freedom Crow. 2017. *Yasuhati*. Freem Inc., Nishinomiya, Japan.
31. Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. 1995. *Design Patterns: Elements of Reusable Object-oriented Software*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
32. Peter Gorniak and Deb Roy. 2005. Probabilistic Grounding of Situated Speech Using Plan Recognition and Reference Resolution. In *Proceedings of the 7th International Conference on Multimodal Interfaces* (ICMI '05), 138–143. https://doi.org/10.1145/1088463.1088489
33. 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 *Proceedings of the 13th IFIP TC 13*

International Conference on Human-computer Interaction - Volume Part I (INTERACT'11), 11–29.

34. Harmonix. 2004. *Karaoke Revolution Volume 2*. Konami, Tokyo.
35. Kenneth Hullett and Jim Whitehead. 2010. Design Patterns in FPS Levels. In *Proceedings of the Fifth International Conference on the Foundations of Digital Games* (FDG '10), 78–85. https://doi.org/10.1145/1822348.1822359
36. Takeo Igarashi and John F. Hughes. 2001. Voice As Sound: Using Non-verbal Voice Input for Interactive Control. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology* (UIST '01), 155–156. https://doi.org/10.1145/502348.502372
37. Indomitus Games. 2015. *In Verbis Virtus*. Indomitus Games, Borgo a Mozzano, Italy.
38. Iridium Studios. 2015. *There Came an Echo*. Iridium Studios, Los Angeles.
39. W. Lewis Johnson and Andre Valente. 2009. Tactical Language and Culture Training Systems: Using AI to Teach Foreign Languages and Cultures. *AI Magazine* 30, 2: 72. https://doi.org/10.1609/aimag.v30i2.2240
40. Joyce Hakansson Associates. 1983. *I'm Hiding*. Milton Bradley, East Longmeadow, MA.
41. Peter H. Kahn, Nathan G. Freier, Takayuki Kanda, Hiroshi Ishiguro, Jolina H. Ruckert, Rachel L. Severson, and Shaun K. Kane. 2008. Design Patterns for Sociality in Human-Robot Interaction. In *Proceedings of the 3rd ACM/IEEE International Conference on Human Robot Interaction* (HRI '08), 97–104. https://doi.org/10.1145/1349822.1349836
42. Anuj Kumar, Pooja Reddy, Anuj Tewari, Rajat Agrawal, and Matthew Kam. 2012. Improving Literacy in Developing Countries Using Speech Recognition-supported Games on Mobile Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '12), 1149–1158. https://doi.org/10.1145/2207676.2208564
43. Labgoo. 2011. *Pah!* Labgoo, Tel Aviv.
44. Petri Lankoski and Staffan Björk. 2007. Gameplay Design Patterns for Believable Non-Player Characters. In *Proceedings of the 2007 DiGRA International Conference: Situated Play* (DiGRA '07).
45. Petri Lankoski, Anja Johansson, Benny Karlsson, Staffan Björk, and Pierangelo Dell'Acqua. 2011. AI Design for Believable Characters via Gameplay Design Patterns. In *Business, Technological and Social Dimensions of Computer Games: Multidisciplinary Developments*. IGI Global, 15. https://doi.org/10.4018/978-1-60960-567-4.ch002
46. Chris Lewis, Noah Wardrip-Fruin, and Jim Whitehead. 2012. Motivational Game Design Patterns of 'Ville Games. In *Proceedings of the International Conference on the Foundations of Digital Games* (FDG '12), 172–179. https://doi.org/10.1145/2282338.2282373
47. Looking West Apps. 2013. *ATC Voice*. Looking West Apps, Denver, CO.
48. Marta Lopes, João Magalhães, and Sofia Cavaco. 2016. A Voice-controlled Serious Game for the Sustained Vowel Exercise. In *Proceedings of the 13th International Conference on Advances in Computer Entertainment Technology* (ACE '16), 32:1–32:6. https://doi.org/10.1145/3001773.3001807
49. Ian McGraw, On Yoshimoto, and Stephanie Seneff. 2009. Speech-enabled Card Games for Incidental Vocabulary Acquisition in a Foreign Language. *Speech Communication* 51, 10: 1006–1023. https://doi.org/10.1016/j.specom.2009.04.011
50. Bria Mears and Jichen Zhu. 2017. Design Patterns for Silent Player Characters in Narrative-driven Games. In *Proceedings of the 12th International Conference on the Foundations of Digital Games* (FDG '17), 59:1–59:4. https://doi.org/10.1145/3102071.3106366
51. David Milam, Lyn Bartram, and Magy Seif El-Nasr. 2012. Design Patterns of Focused Attention. In *Proceedings of the First Workshop on Design Patterns in Games* (DPG '12), 5:1–5:8. https://doi.org/10.1145/2427116.2427121
52. David Milam and Magy Seif El-Nasr. 2010. Design Patterns to Guide Player Movement in 3D Games. In *Proceedings of the 5th ACM SIGGRAPH Symposium on Video Games* (Sandbox '10), 37–42. https://doi.org/10.1145/1836135.1836141
53. Beckett Mufson. 2017. Watch People Play a Hilarious, Frustrating, Voice-Controlled iPhone Game. *VICE Creators*. Retrieved April 14, 2018 from https://creators.vice.com/en_au/article/vvjvk9/hilarious-frustrating-voice-controlled-iphone-game
54. Andrés Adolfo Navarro-Newball, Diego Loaiza, Claudia Oviedo, Andrés Darío Castillo, A. Portilla, Diego Linares, and Gloria Inés Álvarez. 2014. Talking to Teo: Video game supported speech therapy. *Entertainment Computing* 5, 4: 401–412. https://doi.org/10.1016/j.entcom.2014.10.005
55. Neversoft. 2008. *Guitar Hero: World Tour*. Activision, Santa Monica, CA.
56. Nintendo Entertainment Analysis & Development. 2005. *Nintendogs*. Nintendo, Kyoto, Japan.
57. Nintendo Entertainment Analysis & Development. 2007. *The Legend of Zelda: Phantom Hourglass*. Nintendo, Kyoto, Japan.
58. Yue Pan and Erik Stolterman. 2013. Pattern Language and HCI: Expectations and Experiences. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems* (CHI EA '13), 1989–1998. https://doi.org/10.1145/2468356.2468716
59. James R. Parker and John Heerema. 2008. Audio Interaction in Computer Mediated Games. *International Journal of Computer Games Technology* 2008: 1:1–1:8. https://doi.org/10.1155/2008/178923
60. Playground Games. 2014. *Forza Horizon 2*. Microsoft Studios, Redmond, WA.
61. RDI Video Systems. 1984. *Thayer's Quest*. RDI Video Systems, Carlsbad, CA.
62. Gabriel Rivera, Kenneth Hullett, and Jim Whitehead. 2012. Enemy NPC Design Patterns in Shooter Games. In *Proceedings of the First Workshop on Design Patterns in Games* (DPG '12), 6:1–6:8. https://doi.org/10.1145/2427116.2427122
63. Zak Rubin and Sri Kurniawan. 2013. Speech Adventure: Using Speech Recognition for Cleft Speech Therapy. In *Proceedings of the 6th International Conference on Pervasive Technologies Related to Assistive Environments* (PETRA '13), 35:1–35:4. https://doi.org/10.1145/2504335.2504373

64. Serdar Sali, Noah Wardrip-Fruin, Steven Dow, Michael Mateas, Sri Kurniawan, Aaron A. Reed, and Ronald Liu. 2010. Playing with Words: From Intuition to Evaluation of Game Dialogue Interfaces. In *Proceedings of the Fifth International Conference on the Foundations of Digital Games* (FDG '10), 179–186. https://doi.org/10.1145/1822348.1822372
65. Katarina Segerståhl and Timo Jokela. 2006. Usability of Interaction Patterns. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems* (CHI EA '06), 1301–1306. https://doi.org/10.1145/1125451.1125693
66. Ben Shneiderman. 2000. The Limits of Speech Recognition. *Communications of the ACM 43*, 63–65.
67. Gillian Smith, Ryan Anderson, Brian Kopleck, Zach Lindblad, Lauren Scott, Adam Wardell, Jim Whitehead, and Michael Mateas. 2011. Situating Quests: Design Patterns for Quest and Level Design in Role-Playing Games. In *Interactive Storytelling* (Lecture Notes in Computer Science), 326–329. https://doi.org/10.1007/978-3-642-25289-1_40
68. 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. 213–220. https://doi.org/10.1145/1168987.1169023
69. Chek Tien Tan, Andrew Johnston, Kirrie Ballard, Samuel Ferguson, and Dharani Perera-Schulz. 2013. sPeAK-MAN: Towards Popular Gameplay for Speech Therapy. In *Proceedings of The 9th Australasian Conference on Interactive Entertainment: Matters of Life and Death* (IE '13), 28:1–28:4. https://doi.org/10.1145/2513002.2513022
70. Zachary O. Toups, William A. Hamilton, and Sultan A. Alharthi. 2016. Playing at Planning: Game Design Patterns from Disaster Response Practice. In *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play* (CHI PLAY '16), 362–375. https://doi.org/10.1145/2967934.2968089
71. 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. In *Proceedings of the 10th International Conference on the Foundations of Digital Games 2015 (FDG 2015)*, 8. Retrieved from http://www.fdg2015.org/papers/fdg2015_paper_23.pdf
72. Paul Turowski and Simon Hutchinson. 2017. *Plurality Spring*.
73. Ubisoft Montreal. 2006. *Rainbow Six: Vegas*. Ubisoft, Paris.
74. Ubisoft Shanghai. 2008. *Tom Clancy's EndWar*. Ubisoft, Paris.
75. Eduardo Velloso and Marcus Carter. 2016. The Emergence of EyePlay: A Survey of Eye Interaction in Games. In *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play* (CHI PLAY '16), 171–185. https://doi.org/10.1145/2967934.2968084
76. 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* (CHI PLAY '14), 395–398. https://doi.org/10.1145/2658537.2662993
77. Vivarium. 1999. *Seaman*. Sega, Tokyo.
78. Greg Wadley, Marcus Carter, and Martin Gibbs. 2015. Voice in Virtual Worlds: The Design, Use, and Influence of Voice Chat in Online Play. *Human–Computer Interaction* 30, 3–4: 336–365. https://doi.org/10.1080/07370024.2014.987346
79. Greg Wadley and Nicolas Ducheneaut. 2009. The 'Out-of-Avatar Experience': Object-focused Collaboration in Second Life. In *Proceedings of the 11th European Conference on Computer Supported Cooperative Work* (ECSCW 2009), 323–342. https://doi.org/10.1007/978-1-84882-854-4_19
80. José P. Zagal, Staffan Björk, and Chris Lewis. 2013. Dark Patterns in the Design of Games. In *Proceedings of the 8th International Conference on the Foundations of Digital Games (FDG 2013)*, 39–46.