Proceedings of the Fourteenth Artificial Intelligence and
Interactive Digital Entertainment Conference (AIIDE 2018)

CADI – A Conversational Assistive Design Interface for Discovering Pong Variants

Afshin Mobramaein, Morteza Behrooz, Jim Whitehead
University of California, Santa Cruz
mobramaein@soe.ucsc.edu, morteza@ucsc.edu, ejw@soe.ucsc.edu

Abstract

Mixed-initiative PCG systems provide a way to leverage the expressive power of algorithmic techniques for content generation in a manner that lowers the technical barrier for content creators. While these tools are a proof of concept of how PCG systems can aide aspiring designers reach their vision, there are issues pertaining capturing designer intent, and interface complexity. In this paper we introduce CADI (Conversational Assistive Design Interface) a mixed initiative PCG system for creating variations of the game Pong that use conversational natural language to explore the design space of Pong variations. We provide a motivation for the creation of CADI, and discuss the implementation and design decisions taken to address the issues of designer intent and interface complexity in mixed-initiative PCG systems.

Introduction

The field of Procedural Content Generation (PCG) has long focused on utilizing algorithmic techniques to augment and automate game design processes such as asset creation, level design, and rule design. These techniques can be seen as an aide to aspiring game designers that want to create an artifact (i.e. a game, level, or asset) but otherwise might be limited by a lack of technical or artistic knowledge. Of special interest are mixed-initiative approaches to PCG, systems that co-create artifacts with a human user in order to both leverage the expressive power of algorithmic techniques and human designer intent to generate interesting game content.

Systems such as Tanagra (Smith, Whitehead, and Mateas 2010), and Sentient Sketchbook (Liapis, Yannakakis, and Togelius 2013) are examples of mixed-initiative PCG techniques that can be used to create levels for platformer and strategy games from human designer constraints. More recent approaches such as casual creators (Compton and Mateas 2015) explore mixed-initiative design tools as a way of democratizing content creation by stressing ease of use, accessibility, and fun in their design. This philosophy can be seen in the implementation of modern mixed-initiative PCG systems such as Cillr and Wevva (Nelson et al. 2016; Powley et al. 2017). Designers use these tools to create mobile games on-device by manipulating a series of design space parameters based on the concept of fluidic games (Colton et al. 2018; Nelson et al. 2017). These systems are an example of how mixed-initiative design tools can aide aspiring designers to rapidly prototype ideas for games and create novel and interesting content. They do so without a high entry barrier in terms of technical knowledge. While mixed-initiative tools provide enhanced creative support to designers compared to traditional game development tools, some challenges have been identified in recent work (Mobramaein, Whitehead, and Chakraborttii 2018), namely the lack of a model for designer intent, and issues with user interface (UI) complexity.

The issue of designer intent can be seen in the Tanagra platformer level generator, which can only provide computational support for the placement of platforms within the game world. However, the system is not capable of understanding whether a level fits a qualitative description of what the designer wants to capture in the game. If a designer wanted to make a level that is “energetic” or “warm” the system is not capable of creating an artifact that fits such description autonomously. Only a human designer, working with Tanagra, could create such a design User interface complexity simplification is another area of opportunity for mixed-initiative design tools. Nelson et al. (2017) mentions difficulties designers faced with Cillr’s mobile user interface due to the high dimensionality of the design space in the tool, and an occasional apparent lack of feedback from a parameter change by the user.

In earlier work, we propose using conversational interfaces as a way of tackling the dual challenges of capturing designer intent and reducing user interface complexity in mixed-initiative systems (Mobramaein et al. 2018). Due to the broad range of expressive words available in human languages, natural language input can be used to capture designer intent via the use of qualitative terms about the qualities of generated artifacts. In addition, the conversational nature of such interfaces might lead to a novel exploration of a design space via the successive use of expressive natural language terms. This recreates a process similar to collaborative prototyping in which the user directs a design system iteratively along a series of multiple features in the design space of the tool, rather than navigating the design space one parameter at a time. This type of interaction might be able to provide an alternative to the complex user interfaces of tools like Cillr by only providing a simplified input system that can work from either voice or text input. While natural language interfaces address some of the issues present in mixed initiative systems, they are not without their own challenges. Such issues include: frustration with long interactions with a conversational agent, referred as interaction attrition by Mobramaein et al. (2018), finding an initial solution to present the user as an acceptable starting point for the exploration of a design space, and workflow issues that arise from the linear interaction model of conversational agents compared to the more freeform exploration afforded by slider-based UIs in mixed-initiative systems.

Motivation

We believe that implementing a conversational interface for mixed-initiative PCG systems can provide previously unexplored modes of creative support. One of these modes is natural language input (afforded via text or voice) to allow the exploration of the design space using qualitative and affective terms. One example of this kind of interaction is a designer requesting a "zen game" or an "aggressive ball" rather than manipulating a series of sliders in a traditional GUI based tool. These kinds of interactions might allow a designer to explore the design space of the tool in a way that understands their intent for the artifacts they look to create. In addition, exploration of the design space using qualitative and affective terms provides an interaction model that suits an operationalization of game design theories such as game feel (Swink 2008).

Game feel emphasizes the importance of affective experience in game design and focuses on controls, game world design, and aesthetic polish. As such, we can map affective natural language qualifiers such as cold, frenzied, and vibrant as a series of compound features in a parametric design space inspired by Colton et al.'s (2018) concept of emergent features defined as "combinations of features that produce novel emergent effects". One could imagine an affective qualifier as an emergent-like compound feature that combines parameters that modify several aspects of the game at once. For example, a compound feature for "cold" could be a blue-hued color palette, slower moving character speeds, and white colored particle effects. This mapping of affective qualifiers to compound features in mixed-initiative tools could provide us with an operationalization of game feel that can be applied towards computational models of affective experience in PCG.

Exploring a parametric design space using affective qualifiers raises a series of research questions to be answered. How do we provide meaningful feedback to the user as they explore the design space of our mixed-initiative tool? Nelson et al. (2017) mention the occasional cases of user frustration while using Cillr when they modify a parameter slider in the tool, which results in an apparent lack of change in the artifact. We believe that by using compound features tied to a wide combination of game parameters spanning across graphics, physics, and sound, we can provide the user with meaningful feedback on where in the design space they find themselves in, and then take decisions that can lead them to their expected resultant artifact. This can be achieved by applying transformations to the parameters and visualizing the resultant artifact in real time while they interact with the conversational interface.

Another research question concerns the effectiveness of matching natural language input with designer intent. This question applies to understanding the quantitative nature of moves within parameter space to provide a meaningful exploration of the design space, as well as providing the user with a series of mappings of affective qualifiers that are able to cover a wide range of words. The question of understanding the quantitative nature of navigating the design space is important, since commands such as "make the game more intense" or "make the ball less quirky" can be interpreted in a wide variety of ways. With that in mind, it is important to develop an initial strategy of how granular the movement in the design space is, and also a mechanism to adjust the granularity of the movement according to user intent.

Finally, there is the question of how to handle the ambiguity of natural language input for qualitative terms, so user intentions can be understood as accurately as possible. For this, a classification of terms mapped to a series of compound features and parameters in the design space needs to be authored. This affords a flexible approach for grouping terms with similar features that can represent a similar concept. For example, terms such as warm, hot, and heated can be mapped to a feature controlling the palette of our game into a series of values with a red hue in the HSV color space, and to particle systems with a palette in the same tonality. Such a representation might help us understand basic models of polish for PCG as well as to provide an alternative to the parameter-value exploration of a design space.

To explore these design issues, we now introduce CADI, the Conversational Assistive Design Interface, a conversational interface-based system that explores the design space of variations for the game Pong (Alcorn 1972).

Expected User Interaction Modalities

In the design of CADI, we envision two different interaction modalities for how the user can explore the fluidic game design space of Pong variations. These two modalities are called collaborative and autonomous modes. In both modalities, the back-end design space exploration and game generation aspects are the same. The crucial difference rests on the amount of decisions the user takes while exploring the design space of Pong variants and its reflection in the user interface for CADI. This section explores both modalities in the system and provides an explanation of the main interaction loop for our system. Figure 1 shows the interaction loop for CADI.

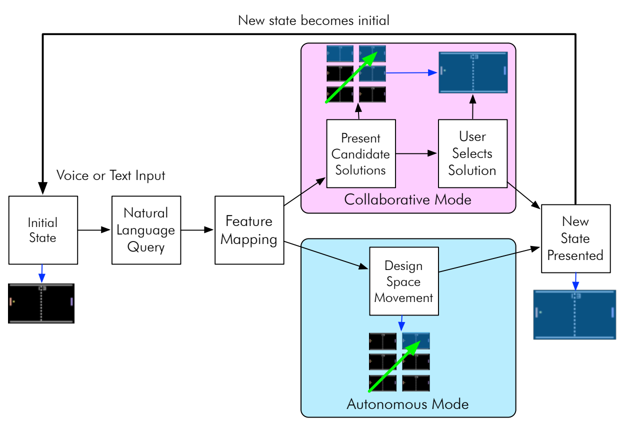


Fig 1. An interaction loop diagram for CADI.

Our interaction loop in the system is comprised by the following steps:

1. The user is provided with an initial base game within our design space. This can be either a vanilla version of Pong faithful to its original design, or a random point in the design space. The user can also select one of these initial designs by using a button in the UI.
2. After being presented with a first point in the design space as a starter, the user can start exploring the design space by using natural language requests either by voice input, or by text input. Some example queries are "make the game more aggressive", "make the paddles heavier", "make the ball red".
3. The queries are passed to our natural language understanding back end, and the quantifier (i.e. less, more, way, quite…), agents (i.e. game, ball, paddle), and qualifier (i.e., vibrant, bouncy, warm) are extracted. These are stored as a JSON object to be passed to our game generation system.
4. The game generation system receives the JSONified request and maps qualifiers to features (i.e. vibrant = {palette_saturation, n_particles}), determines the direction in which the design space is explored (positive, negative) from the quantifier and then moves in the design space on the selected features.
5. The new design variation is presented to the user in a window inside the system's UI.
6. The user continues to make requests until finding a solution suitable to their original intent.

While the interaction loop is essentially the same for both autonomous and collaborative mode, the game generation step differs slightly. In autonomous mode the system moves in the design space, choosing and presenting only one solution for the user. In collaborative mode the system moves in the design space and chooses a series of neighboring points, which are presented to the user. These neighboring points are determined from the current game state by applying a small amount of noise to a randomly chosen sub feature that composes the compound feature that is being modified. This provides the designer with multiple choices among different aesthetic variations of the same design.

System Architecture and Implementation

There are seven different components that make up the architecture of CADI. We implemented five different components, and two components were provided by using operating system features (speech recognition/dictation), and a RESTful API (colourlovers.com) that provides a color palette from a natural language tag. The following figure illustrates the pipeline architecture for CADI.

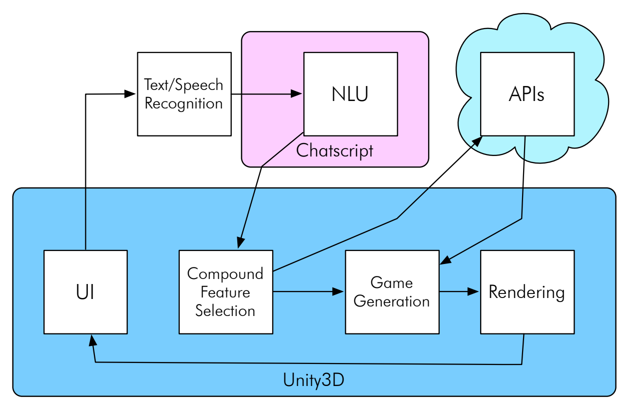


Fig 2. A pipeline architecture diagram for CADI.

CADI was implemented by using the Unity3D engine for the UI, feature selection, game generation and rendering, while, Chatscript (Wilcox 2010) handles the natural language understanding of queries.

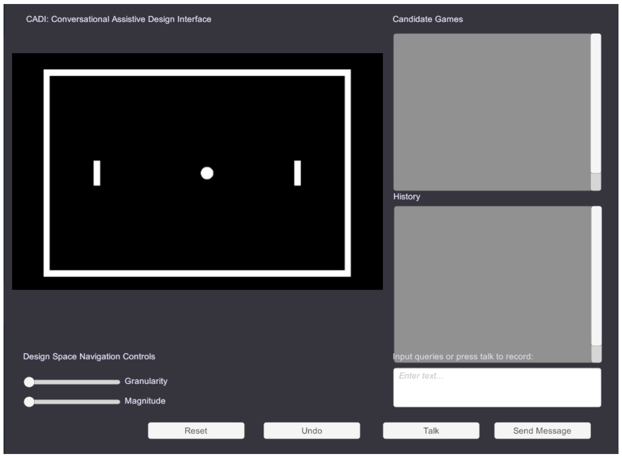


Fig 3. A screenshot of the main UI in CADI

While designing and implementing CADI, several design decisions were taken in order to address the research questions and concerns introduced in our motivation in this paper.

The first design decision was choosing a design space for a conversational based mixed-initiative system. We informed our decision based on the work of Mobramaein et al. (2018) about using Pong variations for exploring conversational interfaces in mixed-initiative PCG systems. This justification comes from the small number of agents in the game (ball, paddles, wall), simple collision based interactions, and high expressivity of the design space. This high expressivity can be seen in game variations such as Video Olympics (Decuir 1977) which uses the elements of Pong to create mechanical variations that reflect different sports, and games such as Pongs (Barr 2012) that explore the design space of pong at a sub textual level. In addition, we utilize the concept of fluidic games (Nelson et al. 2017) as our inspiration to translate the design space of Pong variations into a parameter-vector space over which we can map affective qualifiers from natural language into a series of compound features over which to explore the design space in. We followed Colton et al.'s (2018) methodology in the design process of building our parameter space for Pong variations, and the resultant emergent-like compound features that arise from the concept.

With this justification, we built our parameter space in terms of core features of the games such as the number of agents (balls, paddles) in each variation, different arena shapes, and spatial features (position, rotation) for each agent. Furthermore, parameters were built from the affordances given by the Unity3D engine such as physics parameters for rigidbodies and features specific to trail and particle systems embedded in the game. Finally, aesthetic features related to the game were defined in our implementation for CADI in terms of color palettes, particle systems for collisions, and trail renderers for the agents in the game. This resulted in 29 core parameters for the fluidic game design space of Pong variations.

After designing our core parameters, we proceeded to build our map of compound features to natural language qualitative terms, so as to addresses our research question of how to reason over designer intent in mixed-initiative PCG systems. For this purpose, we opted to author a mapping of grouped natural language concepts to compound features in the game. Our first step was to group words into a series of concepts that can be mapped to a feature in the game. For example, the concept for speed captures words such as fast, slow, quick, lethargic, amongst others. These concepts let us cover a wide number of possible scenarios in which the user wants to modify a feature within our design space. The following table shows a sample of word to compound feature mappings in CADI.

Compound Feature	**Word Concepts**	**Core Feature Combinations**
Vibrant	Colorful, vibrant,	Color palette + Particle FX
Frantic	Frantic, frenzied, hyper	Number of balls, ball speed, paddle speed, trails
Unfair	Asymmetric, unfair, unjust	Paddle size, goal zone size (for only one player)

Table 1. A sample mapping of words to compound features and their associated core features in CADI.

We then authored a series of patterns in Chatscript that map to natural language queries about the game. These patterns can catch queries related to compound features such as "make the game more aggressive" or "make the ball less lethargic". In addition, we added patterns that let the user directly control the core parameters of our design space. An example of a direct control query would be "make the ball red". Our decision to add direct control queries stems from a necessity to cover a wide range of interactions that would be expected from a traditional GUI based mixed-initiative system. By allowing both control over compound features with qualitative terms, and direct control of core parameters in our design space, the user can explore the design space in a manner that reflects their intent to the best of their abilities.

Having designed our NLU component in Chatscript, we were informed by Colton et al.'s (2018) methodology for creating the emergent-like compound features in our game. We built the compound features as a combination of features over which CADI moves in the design space of Pong variations by building a series of example games that encapsulate the meaning of a compound feature. For example, when creating the compound feature for vibrancy we explored the design space by creating example games that capture the intent of the compound feature. After creating the game, we compared our example to an initial solution (original Pong) and determined which parameters changed in the game to capture the concept of vibrancy. In this case, the features modified were the color palette for the game, an application of extra saturation in the HSV color space for the palette, and the application of particle effects to the agents. This process resulted in the creation of 20 different compound features that map to the grouped concepts in our NLU module. Figure 4 shows some example games generated by CADI. A video showing a demo of how these examples is provided in the footnote.[1]

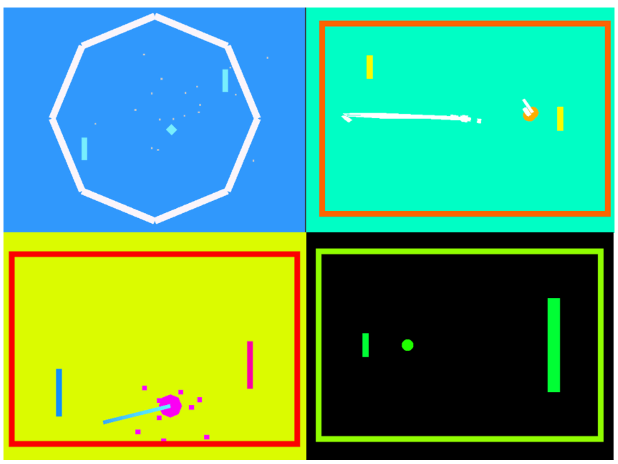


Fig 4. A sample of four different games created by CADI.

After defining our parametric design space, a natural language representation, and a series of compound features that map to natural language concepts, we addressed the research question of how to translate natural language queries into a quantitative movement within the design space of CADI. This is important to address since natural language queries such as "make the ball go faster" or "make the game more colorful" can be ambiguous and the definition of faster or more colorful can vary between users. With this in mind we opted for a simplistic approach that covers a wide range of possibilities of moving along the design space of our system. This was achieved by moving in "steps" along the parameter values for each compound feature. These steps are calculated by the following formulas:

$$\text{step} = \text{m}\left(\text{q} * \frac{|\max(\text{param}) - \min(\text{param})|}{\text{granularity}}\right)$$

$$\text{q} = \begin{cases} 1 \text{ if quantifier} = \text{positive} \\ -1 \text{ if quantifier} = \text{negative} \end{cases}$$

$$\text{m} = \begin{cases} 1 \text{ if not activated} \\ \text{m} \in \mathbb{Z}: \text{m} \in [2,10] \text{ if activated} \end{cases}$$

The steps are calculated by taking the maximum and minimum for each parameter that is modified, and then divided over a user defined granularity parameter that controls how much CADI moves within the design space. The magnitude and direction of our steps are calculated by the values q (for quantifier) and m (for magnifier). The quantifier is extracted from the natural language query in our NLU component with words like "more" meaning a positive movement in the design space, and "less" meaning a negative movement. The magnifier is a combination of a user set parameter and extraction from a natural language query. Phrases such as "even more" and "way less" are processed by our NLU component as a trigger for activating the magnifier variable in our step definition. The values for magnifier and granularity are available to the user within CADI's UI as a couple of sliders. We believe that by letting the user control the amount CADI moves within the design space affords a fine grained control of the design space navigation that can suit the user's needs to the best of their ability. While this scheme is not ideal in the sense that it provides a fully autonomous way of understanding of how to map user queries to a magnitude and direction in the design space, we consider it a first step that might let us understand what users expect when providing natural language queries as movements in the design space of our system. Finally, in the next figure we provide an example of how the user interacts with CADI and the internal steps on how each solution is generated in terms of natural language queries, and feature transformations.

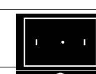
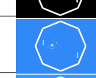
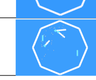
State	Natural Language Query	Compound Feature	Feature Transformation	Game
Initial	none	none	none	
Solution 1	"make the arena rounder"	round	agent: arena feature: n_sides(arena) quantifier: positive	
Solution 2	"make the game look cold"	cold	agent: game feature: color palette (from API) + particle effects quantifier: positive	
Solution 3	"make the ball less round"	round	agent: ball feature: n_sides(ball) quantifier: negative	
Solution 4	"make the game more frantic"	frantic	agent: game feature: n_balls + speed(ball) + speed(paddle) +trail_fx(ball) quantifier: positive	

Fig 5. An example creation for games in CADI.

[1]http:/www.mobramaein.com/CADI

Related Work

Several works in the fields of automated game design (AGD) and mixed-initiative (MI) PCG informed the design and implementation of CADI.

Within the field of AGD, systems like Nelson and Mateas' (2008) Interactive Game-Design Assistant and Game-O-Matic by Treanor et al. (2012) are examples of how a natural language representation of a concept can be used to generate Warioware Inc. (Takeuchi, Abe, and Takahashi 2003) style micro games. In addition, systems such as ANGELINA-3 by Cook, Colton, and Pease (2012) use natural language input to extract a context over which to apply a transformation over the aesthetic design space of a generated game. These systems provide examples of how natural language can be used to create the context of a generated game (Game-O-Matic and Interactive Game-Design Assistant) and how design space movements derive from natural language input (ANGELINA-3).

In the field of MI-PCG there are several example systems that showcase the interaction between a designer and a computational agent to create novel game content artifacts. For example, Sentient Sketchbook (Liapis, Yannakakis, and Togelius 2013) co-create strategy game maps based on a user provided sketch of the artifact. It uses novelty search to explore the design space of potential strategy game maps and presents potential solutions that fit the user's constraints such that a final artifact can be created. This system is of importance to our work as it provides an example of how MI-PCG systems operate over an abstracted representation of the final artifact, in this case a sketch, for the computational agent behind it to create a meaningful artifact that reflects the constraints of the sketch. As such, we model our part of our co-creation (collaborative mode) loop inspired on the interaction loop of this system. In addition, the UI developed in Sentient Sketchbook provided guidelines for CADI's implementation.

In addition, Ropossum (Shaker, Shaker, and Togelius 2013) and Evolutionary Dungeon Designer (Baldwin et al. 2017) provide examples of how evolutionary computation can be used in mixed-initiative design for puzzle and dungeon level design by providing design assistance for playability and providing new solutions based on user constraints. At the intersection of AGD and MI-PCG, there are systems such as Cillr and Wevva (Nelson et al. 2016; Powley et al. 2017) that provide tools that allow designers to co-create mobile mini games on-device. These tools provide an exploration of a fluidic game design space by allowing users to modify parameters in the design space and using an AI agent to evaluate the solutions they explore.

Outside of the domain of games and PCG, there are systems that enable collaboration through natural language between a computational agent and a human user. For example, Firedrop.ai's Sacha (2018) is a chatbot based system that allows the creation of responsive websites from natural language input. In the field of human-robot interaction (HRI) there are systems that utilize spoken dialogue for collaboration between a mobile robot and humans (Kulyukin 2004) similar to the voice input modalities embedded into CADI that allow collaboration between a designer and our system through spoken inputs.

Conclusions and Future Work

We introduce CADI a conversational interface based mixed-initiative PCG system that is capable of exploring the fluidic game design space of Pong variations. We address a series of research questions arising from the implementation of a conversational interface within a mixed-initiative PCG system. These questions pertain to issues like understanding user intent in both quantitative and qualitative terms, as well as providing efficient interaction modalities in our system. We discuss the methodologies and design decisions taken to address our research questions, as well as the details of the implementation of CADI in terms of its design space, knowledge representation, and architecture.

We believe CADI is a first step towards an exploration of different user modalities in mixed-initiative PCG systems. While this implementation addresses some initial questions on how to design a conversational system, and about understanding a design space in qualitative terms through the use of emergent-like compound features, we believe that there is a potential for exploring these issues further. Further work in this project includes an evaluation of CADI compared to a traditional GUI based mixed-initiative system, and an analysis of whether utilizing natural language as a driver for design space exploration addresses the research questions posed in this paper.

Future directions in this field could involve addressing explainability and transparency in conversational driven mixed-initiative systems informed by Zhu et al.'s work (2018) about explainable AI for designers (XAID). In addition, new techniques based on data-driven approaches such as utilizing and transforming open data from the internet (Barros et al. 2018) or the application of neural style-transfer techniques to in-game assets (Colton 2017) could add new depths to this work.

References

Alcorn A. 1972.; Pong (Game). *Atari 2600*: Atari.

Baldwin A., S. Dahlskog, J. M. Font, and J. Holmberg. 2017. "Mixed-Initiative Procedural Generation of Dungeons Using

Game Design Patterns.” In *2017 IEEE Conference on Computational Intelligence and Games (CIG)*, 25–32.

Barr P. 2012.; Pongs (Game). *Pippinbarr.com*

Barros G., et al.; “Data-Driven Design: A Case for Maximalist Game Design.” *arXiv [cs.HC]*, 30 May 2018, http://arxiv.org/abs/1805.12475. arXiv.

Colton S, et al.; *A Parameter-Space Design Methodology for Casual Creators*. 2018, http://repository.falmouth.ac.uk/2893/.

Colton S; 2017 ML Based Style Transfer for Game Assets, ASYNC 17, https://www.youtube.com/watch?v=M5ojBJhQnRk

Compton K.; Mateas M.; 2015. Casual Creators. In *Proceedings of the Sixth International Conference on Computational Creativity.* Park City, Utah: Association for Computational Creativity

Decuir J.; 1977. Video Olympics (Game). *Atari 2600*: Atari.

Firedrop Ventures; 2018, sacha.ai, https://sacha.ai

Kulyukin, Vladimir; 2004. “Human-Robot Interaction Through Gesture-Free Spoken Dialogue.” *Autonomous Robots* 16 (3): 239–57.

Liapis A.; Yannakakis G.; Togelius J.; 2013. Sentient Sketchbook: Computer-Aided Game Level Authoring. In *Proceedings of the Eighth International Conference on the Foundations of Digital Games*. Chania, Crete, Greece: Society for the Advancement of the Science of Digital Games

Mobramaein, A; Whitehead J; Chakrabortti C; 2018 “Talk to Me About Pong: On Using Conversational Interfaces for Mixed-Initiative Game Design.” *AAAI Spring Symposium Series - The UX of AI*.

Nelson M; Mateas M; 2008. “An Interactive Game-Design Assistant.” In *Proceedings of the 13th International Conference on Intelligent User Interfaces*, 90–98. IUI ’08. New York, NY, USA: ACM.

Nelson M.; et al.; 2016. Mixed-Initiative Approaches to On-Device Mobile Game Design. In *Proceedings of the Mixed Initiative Creative Interfaces workshop at CHI 2016.* San Jose, Calif: Association for Computing Machinery

Nelson M.; et al.; 2017. Fluidic Games in Cultural Contexts. In *Proceedings of the Eighth International Conference on Computational Creativity.* Atlanta, Ga: Association for Computational Creativity

Powley EJ.; et al.; 2017 Wevva*: Democratising Game Design*.,

Shaker, N; Shaker M; and Togelius J; 2013. “Ropossum: An Authoring Tool for Designing, Optimizing and Solving Cut the Rope Levels.” In *AIIDE*. http://www.aaai.org/ocs/index.php/AIIDE/AIIDE13/paper/download/7402/7615.

Smith G.; Whitehead J.; Mateas M.; 2010. Tanagra: A Mixed-initiative Level Design Tool. In *Proceedings of the Fifth International Conference on the Foundations of Digital Games*. Monterey, Calif: Society for the Advancement of the Science of Digital Games

Takeuchi K; Abe G; Takahashi R; 2003. WarioWare Inc.: Mega Microgames! (Game), *Gameboy Advance*: Nintendo R&D

Treanor, M; et al.; 2012. “Game-O-Matic: Generating Videogames That Represent Ideas.” In *Proceedings of the The Third Workshop on Procedural Content Generation in Games*, 11. ACM.

Wilcox B.; 2010, Chatscript http://chatscript.sourceforge.net/

Zhu J; et al.; 2018 "Explainable AI for Designers: A Human-Centered Perspective on Mixed-Initiative Co-Creation" in Proceedings of the IEEE Conference on Computational Intelligence and Games, 2018.