Designer Modeling through Design Style Clustering

Alberto Alvarez, Jose Font, and Julian Togelius, *Member, IEEE*

***Abstract*—We propose modeling designer style in mixed-initiative game content creation tools as archetypical design traces. These design traces are formulated as transitions between design styles; these design styles are in turn found through clustering all intermediate designs along the way to making a complete design. This method is implemented in the Evolutionary Dungeon Designer, a research platform for mixed-initiative systems to create roguelike games. We present results both in the form of design styles for rooms, which can be analyzed to better understand the kind of rooms designed by users, and in the form of archetypical sequences between these rooms. We further discuss how the results here can be used to create style-sensitive suggestions. Such suggestions would allow the system to be one step ahead of the designer, offering suggestions for the next cluster, assuming that the designer will follow one of the archetypical design traces.**

***Index Terms*—Procedural Content Generation, Mixed-Initiative Co-Creativity, Designer Modeling, Unsupervised Learning, Computer Games**

I. Introduction

How can we best build a system that lets a human designer collaborate with procedural content generation (PCG) algorithms in order to create useful and novel game content? Various systems have been proposed that allow for humans and algorithms to share authorship by both editing and critiquing the content being created, in what is called the mixed-initiative paradigm [1], [2]. However, for such collaboration to reach its true potential, there needs to be an understanding between the human designer and the software system about what needs to be designed; ideally even a shared goal.

Reaching such a shared understanding is a hard task even when both collaborators share significant cultural and professional background. When one of the collaborators is a computer program, this task is perhaps AI-complete. But we can take steps towards the goal of shared understanding. One idea is to train a supervised learning model on traces of other collaborative creation session and try to predict the next step the human would take in the design process. The main problem with this is that people are different, and different creators will want to take different design actions in the same state; another problem is what to do in design states that have not been encountered in the training data. To remedy this, it has been proposed to train multiple different models, predicting the next step for different designer "personas" (akin to procedural personas in game-playing [3]). However, for such a procedure to be effective, we need to have a sufficient amount of training data. The more different designer personas there are, the more training data is necessary.

One way of overcoming this problem could be to change the level of abstraction at which design actions are modeled and predicted. Instead of predicting individual edits, one could identify different styles or phases of the artifact being created, and model how a designer moves from one to another. To put this concretely in the context of designing rooms for a Zelda-like dungeon crawler [4], one could classify room styles depending on whether they were enemy onslaughts, complex wall mazes, treasure puzzles, and so on. One could then train models to recognize which types of rooms a user creates in which order. By clustering sequences of styles, we could formulate designer personas as archetypical trajectories through style space, rather than as sequences of individual edits. For example, in the context of creating a dungeon crawler, some designers might start with the outer walls of the rooms and then populate it with NPCs, whereas another type of designer might first sketch the path they would like the player to take from the entrance to the exit and then add parts of the room outside the main path. These designer models could then be combined with search-based or other procedural generation methods to suggest ways of getting to the next design style from the current one.

In this paper we provide a prototype implementation of designer personas as archetypical paths through style space. Through this, we take a step further into modeling designers For this we use the Evolutionary Dungeon Designer (EDD), a research platform for exploring mixed-initiative creation of dungeon crawler content [5], [6]. Data from several dozen users designing game levels with the tool have been used to train the models. Based on this data, we clustered room styles to identify a dozen distinct types of rooms. To understand the typical progress of designers and validate the clustering, we visualize how typical design sessions traverse the various clusters. We also perform frequent sequence mining on the design sessions to find a small handful of designer personas.

II. Background

Player modeling, the ability to recognize general socio-emotional and cognitive/behavioral patterns in players [7], has been appointed by the game research community as an essential process in many aspects of game development, such as designing of new game features, driving marketing and profitability analyses, or as a means to improve PCG and game content adaptation. Player modeling frequently relies on data-driven and ML approaches to create such models out of several sorts of user-generated gameplay data [3], [8]–[11].

Using player data from *Iconoscope*, a freeform creation game for visually depicting semantic concepts, Liapis et al.

A. Alvarez, and J. Font are with the Department of Computer Science and Media Technology (DVMT), Malmö University, Malmö, Sweden (e-mail: alberto.alvarez@mau.se; jose.font@mau.se).

J. Togelius is with the Department of Computer Science and Engineering, New York University, New York, NY 11201 USA (e-mail: julian@togelius.com).

trained and compared several ML algorithms by their ability to predict the appeal of an icon from its visual appearance [8]. Furthermore, Alvarez and Vozaru explored personality-driven agents based on individuals' personalities using the *cibernetic big five model*, evaluating how observers judged and perceived agents using data from their personality test when encountering multiple situations [12].

Moreover, training models on gameplay data from *Tom Clancy's The Division* has also been used to model, and therefore find predictors of player motivation [11], which renders a very valuable tool for understanding the psychological effects of gameplay. Former research followed a similar approach in *Tomb Raider Underworld*, training player models on high-level playing behavior data, identifying four types of players as behavior clusters, which provide relevant information for game testing and mechanic design [10]. Melhart et al. take these approaches one step further by modeling a user's *Theory of Mind* in a human-game agent scenario [9], finding that players' perception of an agent's frustration is more a cognitive process than an affective response.

A. The Player is the Designer

Mixed-initiative co-creativity (MI-CC) [1], is the subset of PCG algorithms where human users and AI systems engage in a constant mutual inspiration loop towards the creation of game content [13]–[17]. Understanding player behavior and experience, as well as predicting the player's motivation and intention is key for mixed-initiative creative tools while aiming to offer in real-time user-tailored procedurally generated content. Nevertheless, the player is the designer in MI-CC, and gameplay data is replaced by a compilation of designer-user actions and AI model reactions over time while both user and model are engaged in a mutually inspired creative process. A fluent MI-CC loop should provide good human understanding and interpretation of the system, as well as accurate user behavior modelling by the system, capable of projecting the user's subsequent design decisions [18].

Shifting towards a designer-centric perspective means that besides focusing on player modeling, it is necessary to focus on modeling the designers. Liapis et al. [19], [20] introduced designer modeling for personalized experiences when using computer-aided design tools, with a focus on the integration of such in automatized and mixed-initiative content creation. The focus is on capturing the designer's style, preferences, goals, intentions, and iterative design process to create representative models of designers. Through these models, designer's and their design process could be understood in-depth, enabling adaptive experiences, further reducing their workload and fostering their creativity.

Furthermore, lack of transparency is a key impediment for the advancement of human-AI systems, being eXplainable AI (XAI) an emergent research field that holds substantial promise for improving model explainability while maintaining high-performance levels [21], [22]. However, explanations should be aligned with the users' understanding to don't hinder the usability of systems, as demonstrated by Nourani et al. [23], who discuss the effects of meaningful and meaningless explanations to users of an AI interactive systems.

Zhu et al. [24] proposed the field of eXplainable AI for Designers (XAID) as a human-centered perspective on MI-CC tools. This work discusses three principles of mixed-initiative, *explainability*, *initiative*, and *domain overlap*, where the latter focuses on the study of the overlapping creative tasks between game designers and black-box PCG systems in mixed-initiative contexts. This work deems of high relevance the inclusion of data-driven and trained artifacts to facilitate a fluent bi-directional communication of the internal mechanisms of such a complex co-creative process in which *the designer provides the vision, the AI provides capabilities, and they merge that into the creation*. Mapping the designer's internal model to the AI's internal model is suggested as a meaningful way for creating a common ground that establishes a shared language that enables such communication. In the same line, Xie et al. [25] explored visualization techniques through an interactive level designer tool called *QUBE* to explain and introduce machine learning principles to game designers.

Moreover, Guzdial et al. [26] discuss the insufficiency of current approaches to PCGML for MI-CC, as well as the need for training on specific datasets of co-creative level design. Guzdial et al. work on the mixed-initiative Morai Maker [27] shows the relevance of exploring the ways designers and AI interact towards co-creation, identifying four human-AI relationships (friend, collaborator, student, and manager), as well as the different ways they impact on the designer-user experience. Our study advocates for the importance of designer modeling through ML as the generation of surrogate models of designer styles by training on existing designer-generated data, aiming for an improvement in quality and diversity in computational creativity and, in particular, MI-CC tools.

B. The Designer Preference Model in EDD

EDD is an MI-CC tool where designers can create dungeons and rooms; meanwhile, a PCG system analyzes their design and proposes generated suggestions to the designer [6], [28]. EDD uses the *Interactive Constrained MAP-Elites* (IC-MAP-Elites) [5], an evolutionary algorithm that combines Constrained MAP-Elites [29] with interactive and continuous evolution.

The work presented in [30] introduced the Designer Preference Model, a data-driven solution that learns from user-generated data in the MI-CC Evolutionary Dungeon Designer. This preference model uses an Artificial Neural Network to model the designer based on the choices she makes while using EDD. Both systems constantly interact and depend on each other, so that the Designer Preference Model learns from the generated and selected elites, and IC-MAP-Elites uses the Designer Preference Model as a surrogate model of the designer to complement the fitness evaluation of new individuals.

This approach's main goal is modeling the user's design style to better assess the tool's procedurally generated content, increasing the user's agency over the generated content without stalling the MI-CC loop [18] or increasing user fatigue with periodical suggestion handpicking [2], [31]. The results showed the need for stability and robustness in the data-driven model, to counterbalance the highly dynamic designer's creative process.

III. Concepts and Definitions

Our work draws from many of the ideas and concepts introduced by Liapis et al. [19], in relation to style, goals, preferences and design processes of designers. Nevertheless, given the interdisciplinary scope of this system, and the multiple concepts discuss throughout the paper, it is essential to have operational definitions on the different terms used.

A. Design Style

There exist many different styles when creating content, especially levels, that designers can create and adapt to accomplish their goals and the experiences they want for players. On a general level, *Design Style* can be analyzed as overarching goals that different designers have when creating a dungeon. For instance, dungeons in games such as Zelda [4] or The Binding of Isaac [32], represent a particular playing style planned by the designer. In the former, low tempo, exploring the dungeon, and secret rooms define the style of the dungeons, whereas in the latter, high tempo, optimizing time and resources, small rooms, and in general high-challenge define the dungeons.

While interesting and relevant to understand the designers' holistic design process and the expected player experience, *Design Style* can also be discussed from an individual room basis. Rooms have their own set of characteristics and styles that can be identified and modeled to understand their design process. Some would prefer to create the architecture of the room first to then create the goals within, whereas others would like to place strategic objectives around and then create the architecture around it or alternating between both. Even with such a division, how to reach those design styles is not straightforward and does not require the same strategy, which also shows the preference and style of individual designers. For instance, if the goal is to create a challenge to reach a door, the designer could create a room with a substantial amount of enemies, or create a concentrated high-challenge in the center of the room, or divide the room into smaller choke areas. Therefore, in this paper, we treat *Design Style* as the style designers follow to create a room, informed by the individual steps each has taken connected to their preferences and goals.

B. Designer's Goals

The designer's goal is defined as the current state of rooms and the set of interactions done in the tool or sequence of steps taken thus far, to reach such a state. Goals by the designer are linked to the addition and strategic placement of enemies and treasures, giving some goal for the player, e.g., forcing the fight with an enemy or allowing the player to avoid the conflict through side paths.

C. System Goals

The system goals are defined as the system's approach to support and foster the work of the designer by providing suggestions aligned with her current design or giving assistance, information, visualization, and measurements when needed. In general, when providing suggestions, the system aims at generating rooms among multiple areas of the generative space, simultaneously providing rooms adapted to the designer's goal and different from it.

D. Shared Goals

The shared goals between the system and the designer are defined as the goals the designer has when creating the dungeon and the individual rooms. Thus, in this paper, the shared goal is set and defined by the designer with her design, and as she develops, adapts, and changes, the system seeks to adjust its goals to support the designer's work. Furthermore, the aim of this paper is to propose a system that is able to identify the designer's current goal and style to adapt further the system's goals to provide a personalized experience.

IV. Room Style Clustering

This paper presents an approach and fundamental steps towards the implementation of designer personas: an analysis of designer style clustering to isolate archetypical paths that can be later be used to build ML surrogate models of archetypal designers. Such models would adapt to the dynamic designer during the mixed-initiative creative process by being placed in the solution space, allowing the designer to traverse such space of models as she drifts through the many dimensions of her creative process.

The proposed system builds on top of EDD's Designer Preference Model and preliminary results [30], expanding it to classify the designers' designs based on clusters developed using previously hand-made design sequences by expert and non-expert designers. Figure 1 illustrates our approach in five sequential stages, from data collection to experimentation and results. The first four stages are explained in the following subsections, whereas Section V shows the experimental results.

A. Data Collection

We conducted two user studies where participants were tasked with freely designing a dungeon in EDD and the rooms that compose it with no further restrictions, using all the available tiles i.e. floor, wall, treasure, enemy, and boss tiles. All participants were introduced to the tool before the design exercise. User-generated data was gathered during the complete design session, creating a new data entry every time the designer edited the dungeon. In total, we had 40 participants, 25 of these (i.e. NYU participants) were industry or academic researchers within the Games and AI field, and the other 15 (i.e. MAU participants) were game design students. This resulted in a diverse dataset composed of 180 unique rooms like the ones depicted in Figure 1, that was pre-processed and clustered in the subsequent stages.

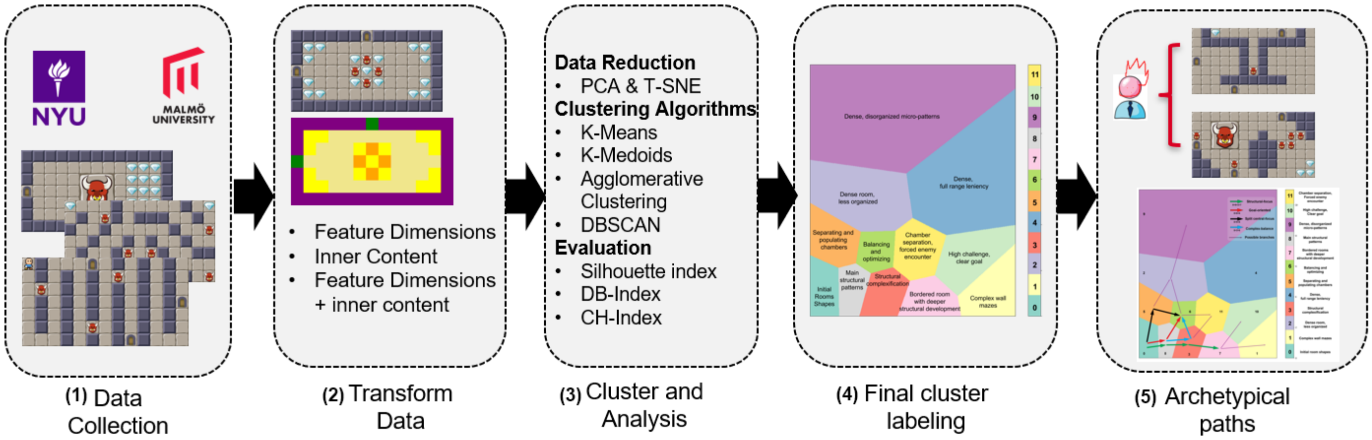


Fig. 1. The stages of the design style clustering development: (1) Data was first collected through two user studies. (2) Then, using the design sequences, the data was processed into five different datasets, one using the room images, a second using the tiles information, and three using tabular information. (3) A data reduction technique was applied to different datasets, and then they were clustered and internally evaluated. (4) The clusters were formed, picked from the best performing methods, and labeled based on the data points within each cluster. The cluster were evaluated by visualizing how a typical design session traverse the various clusters, and K-Means (K=12) was chosen as the final approach. (5) Finally, using this final approach all the sequences were clustered and archetypical paths were identified.

B. Dataset pre-processing

From the 180 unique rooms, we extracted and used the edition sequence of each of the rooms, from their initial design to the more elaborated end-design, to compose a richer dataset that could capture the design process of a designer rather than focusing on the end-point. Through this, we ended up using 8196 data points in our dataset. Moreover, five different copies of the dataset were created to analyze and compare the performance of the clustering stage using the following image pre-processing methods:

1) **Room:** No pre-processing. Room images are fed into the next stage as they were created by the designer, with a resolution of $1300 \times 700 \times 3$, corresponding to width, height, and RGB (3 color channels).
2) **Tiles:** Each room tile type is mapped to a single-color pixel and the rooms are simplified to a pixel-tile based representation, as shown in the second stage of Figure 1. The dimensions are downscaled to $13 \times 7 \times 3$.
3) **Dimensions:** Each room is described by its five IC-MAP-Elites feature dimension values, excluding the similarity scores: LINEARITY, LENIENCY, #MESOPATTERNS, #SPATIALPATTERNS, and SYMMETRY. A complete description of these features can be found in [33].
4) **Inner Content:** Each room is described by 12 values, related to the count, sparsity, and density of the enemy, treasure, floor, and wall tiles contained in it.
5) **Combined:** A combination of the **Dimensions** and **Inner Content** methods.

C. Clustering and Analysis

To run all setups, data reduction algorithms, clustering algorithms, and do the internal evaluation of the clusters, we used scikit-learn machine learning toolset [34]. To obtain the best set of clusters, we ran different setups with the above datasets. The data was reduced to two meaningful dimensions with two different data reduction algorithms, Principal Component Analysis (PCA) and T-Distributed Stochastic Neighbor Embedding (T-SNE). For both data reduction algorithms, we fit the algorithms with each individual dataset, setting to two principal components and in the case of T-SNE using PCA as initializing algorithm, and transforming the data into a new dataset *pca_dataset* and *tsne_dataset* per dataset. Each two-dimensional point in the new datasets represents a step in the sequences described above.

Moreover, all the resulting datasets were then clustered using K-MEANS, K-MEDOIDS, AGGLOMERATIVE CLUSTERING, and DBSCAN. K-Means was initialized using the standard k-means++ implemented in scikit-learn, which initialize all centroids distant from each other. K-Medoids was initialized similarly, using the standard k-medoids++, and tested using the *cosine*, *euclidean*, and *manhattan* distances. Agglomerative clustering is a hierarchical clustering approach using a bottom-up approach implemented in scikit-learn using four different linkage criteria for comparing data points: *Ward*, *Complete*, *Average*, and *Single*. Finally, DBSCAN cluster points based on density separated by low-density areas; thus, DBSCAN automatically finds k based on two parameters, ϵ describing the maximum distance between points and *min_samples* describing the minimum amount of samples within a group to be considered a cluster. K-Means, K-Medoids, and Agglomerative clustering were tested using multiple K values ranging from 3 to 13, and DBSCAN was tested with several ϵ values ranging from 0.3 to 1.0, and *min_samples* ranging from 2 to 9.

Since we lack a labeled dataset (i.e. ground truth) for cluster validation, we evaluated the results from all setups using the internal indices below, as well as manually inspecting the rooms composing the resulting clusters.

- **Silhouette Score:** The Silhouette Score shows how similar a data point is to the cluster it is associated with, through calculating the difference between the $\overline{distance}$

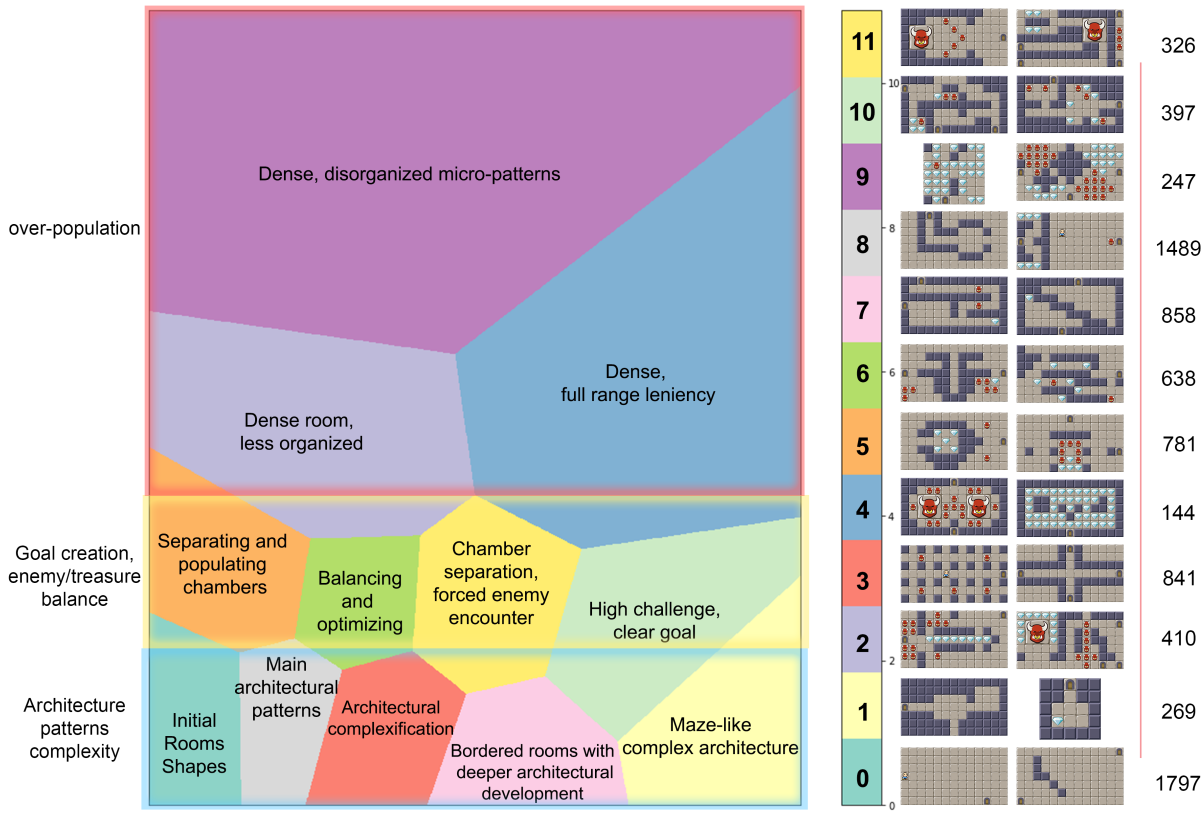


Fig. 2. Best resulting cluster set. K-Means (K=12), using the **Tiles** Dataset. While it scores slightly less in the internal indices that other setups, a qualitative analysis successfully gives us more granularity by subdividing the main bottom clusters, to label and cluster the design process of designers. Sample rooms belonging to each cluster are displayed on the right, next to the total number of rooms in the cluster.

from the point to the points in the nearest cluster and the $distance$ to the points in the actual cluster. The value is bounded from -1 to +1, with values closer to +1 indicating a good separation of the clusters, and closer to -1 meaning that some points might belong to another cluster.

- **Davies-Bouldin Index:** The DB-index is the ratio between the within-cluster distances and between-clusters distances. With this, we can have an insight into the average similarity of clusters with their closest cluster. The value is bounded from 0 to +1, with values closer to 0 relate to clusters that are farther apart from each other and less dispersed, thus, this index is more crucial when we have more dense representations.
- **Calinski-Harabasz Index:** The CH-index is another index related to the density of the clusters and how well separated they are. The score is the ratio between the within-cluster dispersion (compactness) and the between-cluster dispersion (separation). The CH-index is positively unbounded, and the higher the score the better.

D. Cluster Labelling

Table I shows the best performing setups according to their internal indices scores. The clusters in these setups were manually inspected in order to detect the qualitative features that better define them.

When using the **Dimensions** and **Combined** datasets, the clusters do perform good, if not better, in certain indices than when using the **Tiles** dataset. However, when analysing the resulting setups, they were missing a clearer relation between the clustered rooms, which was exacerbated when analysing sequences and paths on these setups, where they missed continuity between clusters.

Conversely, given that we are creating tile-based rooms and dungeons, the features were more representative for the **Tiles** dataset, which when used, generally performed well in the evaluated internal indices, and the produced clusters meaningfully separate the data. Further, as it will be presented in Section V, when clustering sequences and analyzing the cluster path of the designs, there exist a continuity between designs that supports its usability. Figure 2 shows the best-resulting cluster set found among all the experiments run.

In the figure, we have plotted on top of the clusters the labels describing in general, the content that is within them.

TABLE I
BEST PERFORMING SETUPS BASED ON THEIR INTERNAL VALIDATION AND VISUALIZATION OF CLUSTERED DATA POINTS.

Algorithm	Data	K	◊	□	△
K-Means	Tiles-PCA	9	0.43	0.73	9438.233
K-Means	Tiles-PCA	12	0.41	0.77	9436.928
K-Means	Dimensions-PCA	12	0.43	0.73	7738.343
Agglomerative single	Combined-PCA	6	0.51	0.43	38.833
Agglomerative avg.	Dimensions-PCA	6	0.44	0.67	3463.567

◊ Silhouette Score □ Davies Bouldin Index △ Calinski-Harabasz Index

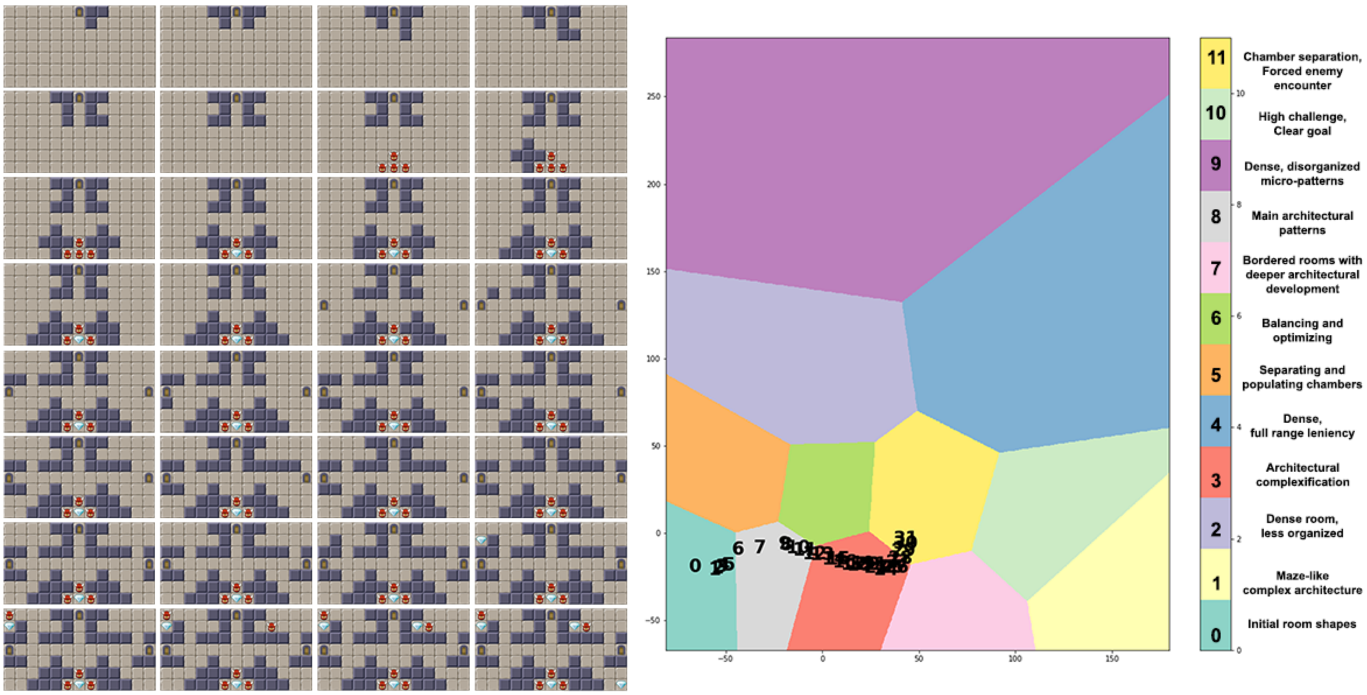


Fig. 3. Example of a step by step edition sequence of a design session and it's clustering. At the left, we present the actual sequence and steps of one of the rooms in the dataset, in a 4×7 grid, starting at the top left with the first edition. At the bottom, it is the actual trajectory of the design in the cluster space. Numbered and in black, it is shown how each step of the design process is clustered by our approach

The following is a description of the clusters and the rooms that were clustered together:

0. Empty-Initial rooms: This cluster relates mostly to the initial designs made by the designers. These designs are from completely empty rooms to initial work-in-progress structures.

1. Maze-like complex architecture: This cluster to the extreme of the architectural patterns complexity layer, relates to more highly-linear, confined and maze-like rooms.

2. Dense, less organized: This cluster contains rooms that still have a certain objective but are moving towards more disorganized distributions of micro-patterns in relation to their density.

3. architectural complexification: This cluster relates mostly to the complexification of wall structures by having dense wall chunks, representative architectural patterns, or symmetrical patterns.

4. Dense, full range leniency: Focusing on density as the other two clusters within the same layer, this cluster relates to rooms that are in the full range of leniency from very rewarding, treasure rooms to very challenging boss rooms.

5. Separating and populating chambers: This cluster relates to the process of separating rooms into distinct chambers, focusing on the center of the room, and starting to populate rooms with enemies and treasures.

6. Balancing and optimizing: This cluster contains a mix between corridors and chambers within rooms with a focus on balancing rooms and optimizing their design towards certain goals.

7. Bordered rooms with deeper architectural development: This cluster relates mostly to rooms with an added wall border by the designer, and where the focus is to shape chambers and develop more visual structures.

8. Main architectural shapes: Similar to other clusters within the same layer, this cluster relates to the development and definition of main architectural patterns that are somewhat symmetric.

9. Dense, disorganized micro-patterns: This cluster clusters the extreme rooms that contain a high density of tiles, other than floor-tiles, without a clear structure or objective for the player.

10. High challenge, clear goal: This cluster relates to well-shaped rooms with clear wall structures and goals, towards more challenge.

11. Chamber separation with forced enemy encounter: This cluster relates to rooms that are in the process of a clear segmentation into corridors and chambers, and that enforce to some extent, enemy encounters for the player.

Furthermore, besides the local relation between clusters, the clusters are implicitly divided in three layers on the Y-axis. From bottom to top, (a) architectural patterns complexity, relating to clusters composed of rooms with clearer or complex shapes done with walls, from empty rooms to mazes. (b) Goal creation, enemy/treasure balance, with clusters comprehending the strategic addition of enemies and treasures to establish objectives in the room for the player. In terms of EDD, these rooms are composed of more meso patterns. And (c), over-population, which relates to clusters filled with less organized and dense rooms where the enemy and treasure addition do not necessarily need to follow any clear objective. Identifying the designer in such layer, and the path they have taken to get there could show meaningful information in the design process. For instance, the intentions of the designer, in what phase of the design process she is at the moment i.e. trying the tool or observing how the tool reacts or scraping her current goal towards a new goal within the room.

V. Designer Personas

Once we created, evaluated, and labeled the clusters, we were able to cluster and visualize the paths of a typical design session. Figure 3 presents an example of the design sessions, where we cluster each step of the design. This sequential process revealed that there is an interesting continuity between clusters, even capturing when a designer probably applied one of the procedural suggestions due to bigger steps in the design style clusters. Further, through this process, we could understand the progress of designers in their design process and represent their trajectory in relation to the traversed clusters rather than individual editions.

A. Unique Trajectories

Using the clusters in Figure 2, we clustered the design session of all the 180 designs and collected the unique trajectories that arose from traversing the various clusters. These unique trajectories varied in the starting point, length, and end-point, however, when analyzing the trajectories we identified common patterns among them. They had a similar shape as the following $Unique = \{0>8>4>7>10\}$, where the first and last element of the sequence are respectively, the starting- and end-points, with all the unique intermediate steps in between.

To gather the common patterns from the trajectories, we applied the Generalized Sequential Pattern (GSP) algorithm, which locates frequent subsequences in the analyzed trajectories. For instance, given three trajectories (a) $\{5>1>3>11>9\}$, (b) $\{5>1>3>11>4$ and (c) $\{0>1>3>11\}$, none of these is a perfect match in its entirety, but GSP can spot that subsequences $\{1>3>11\}$, $\{1>3\}$, $\{3>11\}$, among others, appear with frequency $= 3$.

Furthermore, after doing a preliminary analysis, we identified some steps that we classified as "border designs": steps that are borderline between two clusters. These *border designs* disrupted the sequence pattern mining by creating noise in the unique trajectories, specifically when these *border designs* entered a different cluster for just a few steps. Therefore, we filtered them out by applying a threshold $\theta = 3$, so that all subsequences inside one cluster with less than θ steps are removed from the main sequence. I.e, the sample trajectory $\{0>0>0>0>8>8>8>6>8\}$ turns into $\{0>8\}$ instead of $\{0>8>6>8\}$. Through this, we were able to reduce the noise and the search space, obtaining more meaningful and frequent patterns.

B. Archetypical Paths through Style Space

In Figure 4, we present the archetypical paths, represented as thicker arrows to denote direction, which show the most frequent paths taken by designers either through their whole design process or as the initial meaningful steps. From all the collected unique trajectories, we have identified 4 main archetypical paths, labelled, ARCHITECTURAL-FOCUS, GOAL-ORIENTED, SPLIT CENTRAL-FOCUS, and COMPLEX-BALANCE. In addition, we have numbered each cluster for easier visualization and referencing.

Moreover, in the figure, it can also be observed thinner purple arrows pointing to different clusters from several of the clusters that are part of the main paths. These are *possible branches* presented in the unique trajectories and added based on their frequency. Through these possible branches, the design of an archetypical session, can vary and extended or deviate the final design. Each archetypical path is defined and explained as follows:

1) Architectural-focus: The path followed by this archetype focuses first on designing the architecture of the room with walls. Through this, the design focuses on shaping the visual patterns, chambers, and corridors to give a clear space for adding goals and objectives with enemies and treasures. The sequence is denoted with a green arrow in Figure 4, and following the sequence $\{0>8>3>7\}$.

2) Goal-oriented: Design processes following this archetypical path, create the rooms in a more standard way, combining simpler symmetric wall structures with distributed placement of enemies and treasures. Thus, rather than focusing extensively on an individual part of the room, the rooms have an initial structure and then they are populated with some specific goal-in-mind. The sequence is denoted with a red arrow in Figure 4, and following the sequence $\{0>8>6\}$.

3) Split central-focus: This archetypical path focuses on designing rooms with obstacles placed in the center of the room in the shape of enemies, treasures, or wall structures that clearly split the room into different areas. The design process is less organized than the other archetypes since it searches to achieve the split goal with any of the available tiles. The sequence is denoted with a black arrow in Figure 4, and following the sequence $\{0>5>6\}$.

4) Complex-balance: This archetypical path focuses on building complex symmetric shapes with a clear objective for the player and adapting the spaces with a balance of enemies and treasures. In general, the rooms created following this path are more unique and typically balanced. The sequence is denoted with a blue arrow in Figure 4, and following the sequence $\{8>3>6\}$.

Furthermore, using these archetypical paths, we can then categorize certain clusters as key clusters or being more relevant than others based on their contribution to the paths, their frequency, and their usage. Most of the paths go through or end in cluster 6 ("Balancing and optimizing") and cluster 8 ("Main architectural patterns"), which relate to rooms that have a more explicit mix between corridors and small chambers and more clear architecture. The rooms in those clusters are or shaped as end rooms, as in the case of cluster 6, or architecturally shaped to be "optimized" to a specific goal e.g. a dense bordered room. Similarly, most of the sequences start from cluster 0 ("Initial room shapes"), with 134 out of the 180 designs, which correlates to the type of designs encountered in that clusters. Thus, it is understandable that most of the archetypical paths pass through any of these three clusters.

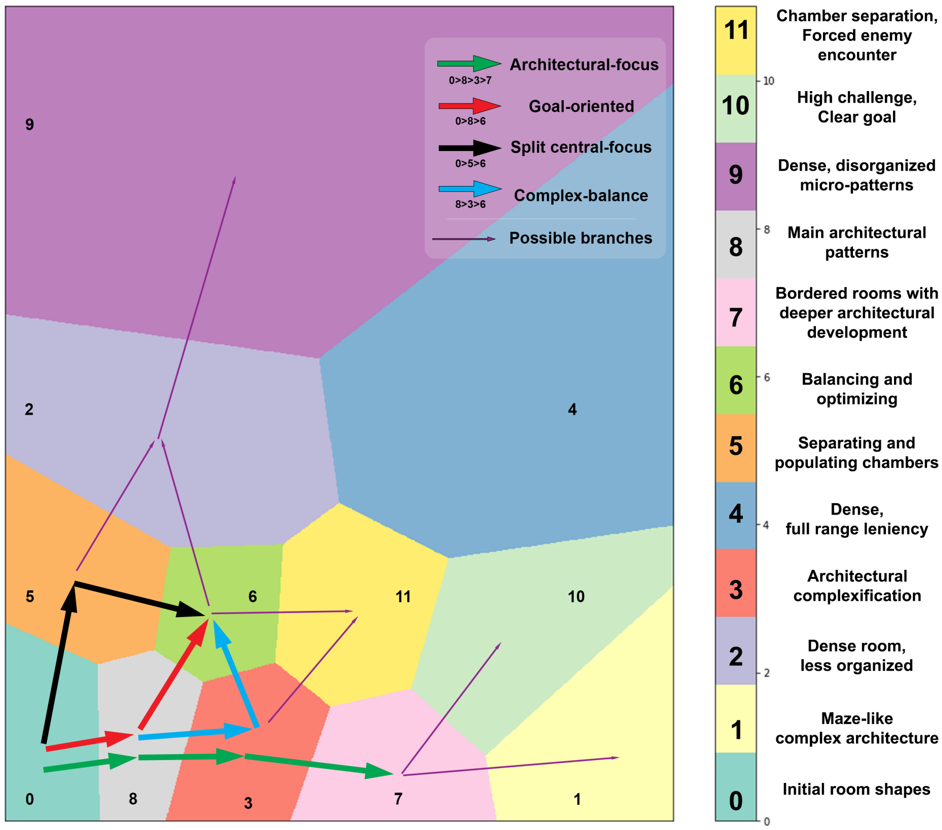


Fig. 4. Final and common designer trajectories. With thick arrows it is presented the archetypical paths, calculated using the frequencies of subsequences from 180 diverse rooms. Each color represent a unique trajectory; with green the ARCHITECTURAL-FOCUS, with red the GOAL-ORIENTED, with black the SPLIT CENTRAL-FOCUS, and with blue the COMPLEX-BALANCE. Finally, thinner purple arrows extending from clusters traversed by the archetypical paths show the multiple possible branches that an archetypical path can deviate or extend to.

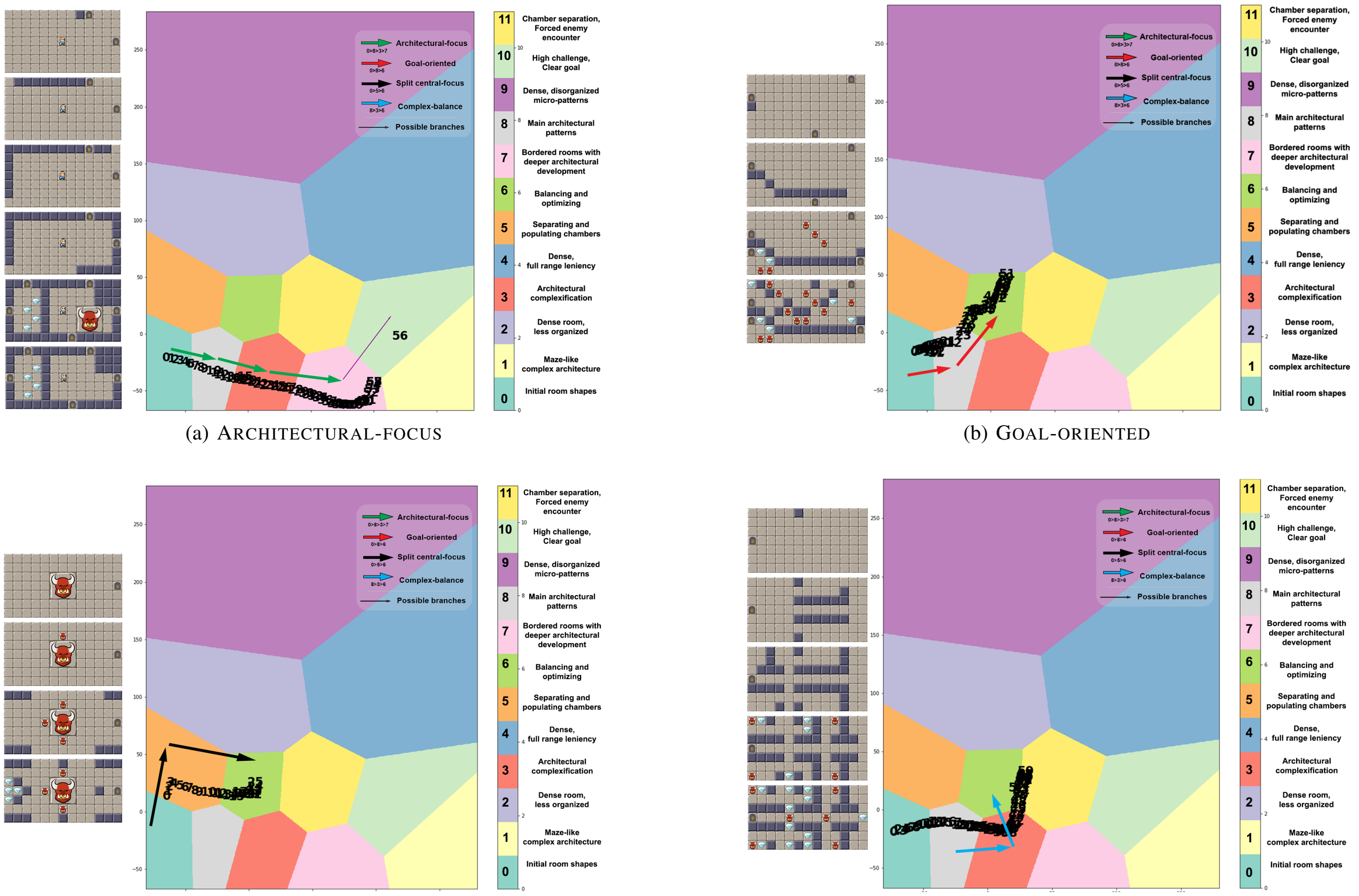
Nevertheless, it is the steps in-between what creates a clear differentiation between the archetypical paths, which is the benefit of observing the design process as a whole in the clustered room style space. For instance, in fig. 4, it can be observed that SPLIT CENTRAL-FOCUS starts in the same cluster as three other paths, and tentatively ends in the same cluster as three other. However, the designs following SPLIT CENTRAL-FOCUS are more different to the other trajectories, since it enters a cluster that is denser with several tile types in principle, and where designers seem to have a clearer goal.

Moreover, in figure 5, we present examples of each of the designer personas by visualizing the sequence of steps done in representative design sessions, showing how these paths would look like in practice. Each visualization of a designer persona has the key design steps to the left, where each image is in a sequence: the first is the first edition of the designer, the last is the final edition, and the in-between represent entering a new room style cluster.

In (a), it is shown the ARCHITECTURAL-FOCUS, where the designer first created the border of the room with a clear chamber division. As the designer adds and subsequently removes the boss, the design jumps to cluster 10, which is one of the possible branches, adding a high challenge. In (b), it is shown the GOAL-ORIENTED, where the designer sketched the main shape of the room followed by alternating between enemies, treasures, and walls to design the goal of the player within the room. In this example, the designer ends the design close to cluster 9, with a disorganized placement of tiles and a less aesthetical room, but forming small choke areas balancing the placement of enemies and treasures.

In (c), it is shown the SPLIT CENTRAL-FOCUS, where the designer directly started by adding a boss in the center of the room and using this as a reference point, shaped the rest of the room. In (d), it is shown the COMPLEX-BALANCE, where the designer focused on creating an uncommon structure and followed by adding enemies and treasures symmetrically, with clear individual areas for the player to approach.

Finally, further analyzing figure 5, it can also be observed an interesting dual tendency of the designers in the archetypical paths. This dual tendency is to either focus on the aesthetic configuration of the room based on what is perceived in the editor exemplified the personas: ARCHITECTURAL-FOCUS and SPLIT CENTRAL-FOCUS, and to focus on the player experience exemplified the personas: GOAL-ORIENTED and COMPLEX-BALANCE. Nevertheless, both are not mutually exclusive, instead this illustrates adequately the dualistic role the designer has when using the tool and designing rooms. That of creating an aesthetically pleasing object as it is seen in the editor, and that of creating an experience.



(a) ARCHITECTURAL-FOCUS

(b) GOAL-ORIENTED

(c) SPLIT CENTRAL-FOCUS

(d) COMPLEX-BALANCE

Fig. 5. Examples of each of the archetypical paths from one of the frequent sequences used to create the clusters. To the left of each subfigure, we present each key step in the trajectory i.e. when the design entered a new cluster. (a) presents the ARCHITECTURAL-FOCUS archetypical path where the focus is firstly on creating the structural design of the rooms; the design process jumps back and forth suddenly to cluster 10 (one of the possible branches) due to the designer adding a boss, and removing it immediately. (b) presents the GOAL-ORIENTED archetypical path where the design focus on a minimal structure complexity and mix between adding structural changes and enemies/treasures. (c) shows the SPLIT CENTRAL-FOCUS archetypical path where, intentionally, the designer creates a center obstacle with a boss and build around it. Finally, (d) presents the COMPLEX-BALANCE archetypical path; the design focuses on building complex, uncommon structures first and then add some goal to it with enemies and treasures, taking advantage of the spaces.

VI. CONCLUSIONS

This paper presents a novel approach and meaningful steps towards designer modeling through an experiment on archetypical design trajectories analysis in an MI-CC environment. Through this, we characterize several representative design styles as designer personas. We have first run and compared several clustering setups to find the best partitioning of the design style using the edition sequences of the collected 180 unique rooms, ending in 8196 data points, and resulting in a set of twelve cohesive, coherent, and meaningful clusters. We have then mapped these 180 design sequences in terms of these clusters, applying frequent sequence mining to find four frequent and unique designer styles, with related common sub-styles. As a result, we have presented a roadmap of design styles over a map of data-driven design clusters.

Designer modeling was proposed as an approach to capture multiple designer's processes to create a better workflow by Liapis et al. [19], and our work draws on many of their ideas, concepts, and goals. Furthermore, a prototype of such was implemented in the sentient sketchbook [20], where it is proposed different approaches to model style, process, and goals based on choice-based evolution and the designer's current design to adapt the provided suggestions accordingly. We propose an alternative route to designer modeling through clustering the design space and the room style based on the collected data. Moreover, we differ in the type of level design, being the sentient sketchbook a tool for strategy games [17], while EDD is a tool for adventure and rogue-like games [33]. These differences strengthen the importance and usefulness of designer modeling and highlight the holistic and generic properties of this designer-centric perspective and its possibilities.

These contributions allow us to better understand, cluster, categorize and isolate designer behavior. This is very valuable for mixed-initiative approaches, where a clear virtual model of the designer's style allows us to better drive the search process for procedurally generating content that is valuable for the designer. Designer personas have the potential to be used in many different scenarios. For instance, as objectives for a search-based approach to enable a more style-sensitive system, to evaluate the fitness of evolutionary generated content or to train PCG agents via Reinforcement Learning [35].

Moreover, recognizing the designers' current style and the path taken so far, which would indicate a possible designer persona, could open the possibility for recognizing their intentions, preferences, and goals. This traced roadmap of designer personas could let a content generator anticipate a designer's next moves without heavy computational cost, just by identifying her current location on the map and offering content suggestions that lie in the most promising clusters to be visited next. Conversely, it could also identify designers who do not follow a certain path, i.e. deviating from the pattern, trying to understand their objective through their design style.

Finally, it is also important to observe the nature of the previous and future rooms created by a designer. Observing the dungeon as a whole, as briefly introduced in section III-A, to understand the designers' intentions and goals when they proceed to create a new room is a promising future step to take with the current system.

REFERENCES

[1] G. N. Yannakakis, A. Liapis, and C. Alexopoulos, "Mixed-initiative co-creativity," in *Proceedings of the 9th Conference on the Foundations of Digital Games*, 2014.

[2] A. Liapis, G. Smith, and N. Shaker, "Mixed-initiative content creation," in *Procedural content generation in games*. Springer, 2016, pp. 195–214.

[3] C. Holmgård, M. C. Green, A. Liapis, and J. Togelius, "Automated playtesting with procedural personas through mcts with evolved heuristics," *IEEE Transactions on Games*, vol. 11, no. 4, pp. 352–362, Dec 2019.

[4] Nintendo R&D4, "The Legend of Zelda," 1986.

[5] A. Alvarez, S. Dahlskog, J. Font, and J. Togelius, "Empowering quality diversity in dungeon design with interactive constrained map-elites," in *2019 IEEE Conference on Games (CoG)*, 2019.

[6] A. Baldwin, S. Dahlskog, J. M. Font, and J. Holmberg, "Towards pattern-based mixed-initiative dungeon generation," in *Proceedings of the 12th International Conference on the Foundations of Digital Games*, ser. FDG '17. New York, NY, USA: ACM, 2017.

[7] R. Thawonmas, J. Togelius, and G. N. Yannakakis, "Artificial general intelligence in games: Where play meets design and user experience," *NII Shonan Meeting No. 130*, 2019.

[8] A. Liapis, D. Gravina, E. Kastbjerg, and G. N. Yannakakis, "Modelling the quality of visual creations in iconoscope," in *Games and Learning Alliance*, A. Liapis, G. N. Yannakakis, M. Gentile, and M. Ninaus, Eds. Cham: Springer International Publishing, 2019, pp. 129–138.

[9] D. Melhart, G. N. Yannakakis, and A. Liapis, "I feel i feel you: A theory of mind experiment in games," *KI-Künstliche Intelligenz*, pp. 1–11, 2020.

[10] A. Drachen, A. Canossa, and G. N. Yannakakis, "Player modeling using self-organization in tomb raider: Underworld," in *2009 IEEE Symposium on Computational Intelligence and Games*, Sep. 2009, pp. 1–8.

[11] D. Melhart, A. Azadvar, A. Canossa, A. Liapis, and G. N. Yannakakis, "Your gameplay says it all: Modelling motivation in tom clancy's the division," in *2019 IEEE Conference on Games (CoG)*, Aug 2019, pp. 1–8.

[12] A. Alvarez and M. Vozaru, "Perceived behaviors of personality-driven agents," *Violence — Perception — Video Games: New Directions in Game Research*, pp. 171–184, 2019.

[13] M. Charity, A. Khalifa, and J. Togelius, "Baba is y'all: Collaborative mixed-initiative level design," *arXiv: 2003.14294*, 2020.

[14] T. Machado, D. Gopstein, A. Nealen, and J. Togelius, "Pitako-recommending game design elements in cicero," in *2019 IEEE Conference on Games (CoG)*. IEEE, 2019.

[15] N. Shaker, M. Shaker, and J. Togelius, "Ropossum: An authoring tool for designing, optimizing and solving cut the rope levels." in *AIIDE*, 2013.

[16] G. Smith, J. Whitehead, and M. Mateas, "Tanagra: Reactive Planning and Constraint Solving for Mixed-Initiative Level Design," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, no. 3, Sep. 2011.

[17] A. Liapis, G. N. Yannakakis, and J. Togelius, "Generating Map Sketches for Strategy Games," in *Proceedings of Applications of Evolutionary Computation*, vol. 7835, LNCS. Springer, 2013.

[18] K. Compton, "Casual creators," PhD Dissertation, University of California Santa Cruz, 2019.

[19] A. Liapis, G. Yannakakis, and J. Togelius, "Designer modeling for personalized game content creation tools," in *Artificial Intelligence and Game Aesthetics - Papers from the 2013 AIIDE Workshop, Technical Report*, vol. WS-13-19. AI Access Foundation, 2013, pp. 11–16.

[20] A. Liapis, G. N. Yannakakis, and J. Togelius, "Designer modeling for sentient sketchbook," in *2014 IEEE Conference on Computational Intelligence and Games*, Aug 2014, pp. 1–8.

[21] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (xai)," *IEEE Access*, vol. 6, pp. 52 138–52 160, 2018.

[22] F. Doshi-Velez and B. Kim, "Considerations for Evaluation and Generalization in Interpretable Machine Learning," in *Explainable and Interpretable Models in Computer Vision and Machine Learning*, H. J. Escalante, S. Escalera, I. Guyon, X. Baró, Y. Güçlütürk, U. Güçlü, and M. van Gerven, Eds. Cham: Springer International Publishing, 2018, pp. 3–17.

[23] S. M. E. D. R. Mahsan Nourani, Samia Kabir, "The Effects of Meaningful and Meaningless Explanations on Trust and Perceived System Accuracy in Intelligent Systems," in *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, 2019.

[24] J. Zhu, A. Liapis, S. Risi, R. Bidarra, and G. M. Youngblood, "Explainable ai for designers: A human-centered perspective on mixed-initiative co-creation," in *2018 IEEE Conference on Computational Intelligence and Games (CIG)*, Aug 2018, pp. 1–8.

[25] J. Xie, C. M. Myers, and J. Zhu, "Interactive visualizer to facilitate game designers in understanding machine learning," in *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019, pp. 1–6.

[26] M. Guzdial, N. Liao, and M. Riedl, "Co-creative level design via machine learning," in *Joint Proceedings of the AIIDE 2018 Workshops co-located with 14th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE 2018), Edmonton, Canada, November 13-14, 2018.*, 2018.

[27] M. Guzdial, N. Liao, J. Chen, S.-Y. Chen, S. Shah, V. Shah, J. Reno, G. Smith, and M. O. Riedl, "Friend, collaborator, student, manager: How design of an ai-driven game level editor affects creators," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019, pp. 1–13.

[28] A. Alvarez, S. Dahlskog, J. Font, J. Holmberg, C. Nolasco, and A. Österman, "Fostering creativity in the mixed-initiative evolutionary dungeon designer," in *Proceedings of the 13th International Conference on the Foundations of Digital Games*, ser. FDG '18, 2018.

[29] A. Khalifa, S. Lee, A. Nealen, and J. Togelius, "Talakat: Bullet hell generation through constrained map-elites," in *Proceedings of The Genetic and Evolutionary Computation Conference*. ACM, 2018.

[30] A. Alvarez and J. Font, "Learning the designer's preferences to drive evolution," in *Proceedings of the 23rd European Conference on the Applications of Evolutionary and bio-inspired Computation*, ser. EvoApplications '20, 2020.

[31] H. Takagi, "Interactive evolutionary computation: fusion of the capabilities of ec optimization and human evaluation," *Proceedings of the IEEE*, vol. 89, no. 9, pp. 1275–1296, 2001.

[32] E. McMillen and F. Himsl, "The Binding of Isaac," 2011.

[33] A. Alvarez, S. Dahlskog, J. Font, and J. Togelius, "Interactive constrained map-elites: Analysis and evaluation of the expressiveness of the feature dimensions," *arXiv: 2003.03377*, 2020.

[34] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[35] A. Khalifa, P. Bontrager, S. Earle, and J. Togelius, "Pcgrl: Procedural content generation via reinforcement learning," 2020.