Proceedings of the Twenty-First AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE 2025)

Designer Difficulties: Visualizing the Possibility Spaces of Dynamic Difficulty Adjustment Systems

Samuel Shields[1], **Oliver Withington**[2], **Edward Melcer**[3]
[1]University of California, Santa Cruz, Santa Cruz, CA
[2]Queen Mary University of London, London, UK
[3]Carleton University, Ottowa, ON
samshiel@ucsc.edu, owithington@hotmail.com, edwardmelcer@cunet.carleton.ca

Abstract

Dynamic Difficulty Adjustment (DDA) systems procedurally tune video games during runtime to deliver a designer-specified, balanced gameplay experience for players. DDA systems have been well studied and deployed in both academic and industry contexts. However, a relatively unexplored challenge of DDA systems is how to assess the diversity of experiences they can deliver to players and whether or not the range of possible DDA actions will satisfy the original game designer's goals. DDA systems are inherently unpredictable in their output due to being designed to react to game-states that are uncertain and ever-changing. The varying scope and unpredictability of DDA systems means human playtesting can be time- and cost-intensive, and automated playtesting may produce misleading results. In this work, we introduce an approach for using expressive range analysis and unsupervised clustering to explore and evaluate gameplay-traces from an in-development DDA system (FighterDDA) for turn-based role-playing game encounters. We find that this is an effective method for assessing designer goals and re-tuning accordingly an in-development system by visualizing and understanding the character of different DDA approaches. Although specific to this system, we believe that there is promise in extending this approach to other genres and DDA platforms in the future.

Code — https://github.com/smshields/FighterDDA

Introduction

Dynamic Game Balancing (DGB) is the process of altering properties of a video game during runtime such that it achieves a designer's desired experience of balance for the player (Andrade et al. 2006). A subcategory of DGB is known as Dynamic Difficulty Adjustment (DDA), which focuses specifically on the tuning of difficulty for one or more players during the game (Baldwin et al. 2013). DDA implementations have a variety of forms such as linear scaling of enemy difficulty (e.g., *The Legend of Zelda: Breath of the Wild* (Nintendo EPD 2017) where enemies spawn with more health and damage output based on the main character's progression; mechanics that aid low-ranked players (e.g., *Mario Kart 8* (EAD 2014) where players in last place receive disproportionally powerful items to help them catch up); or the AI Director pattern (e.g., *Left 4 Dead* (South 2008) where an AI agent assigns enemy spawn rates based on a perceived player affect relating to intensity). DDA systems have been widely used in game design since the early days of the medium, and more recently have become a field of active academic study—with much research focused on how DDA systems should be implemented and how they are perceived by players (Silva, do Nascimento Silva, and Chaimowicz 2017; Reis, Reis, and Lau 2021; Kavanagh et al. 2019; Hunicke 2005).

Despite their prevalence, DDA system tuning and testing is often unstructured, typically relying on designer expertise to identify what methods to use to achieve balance and how they should be implemented and tuned (Jaffe et al. 2012). This is a significant issue due to the large number of parameters and methods that can be adjusted to impact difficulty in a typical game, and due to the unpredictable combinatorial effect that changing multiple parameters could have (Dziedzic and Włodarczyk 2018). In addition, there is evidence that the perception of DDA systems can vary significantly between audiences and players (Mortazavi, Moradi, and Vahabie 2024). In practice, this means that balancing during the game development lifecycle is time-consuming, occurs throughout development, and frequently requires many playtesting cycles to be properly evaluated (Jeon et al. 2023; Kokkinakis et al. 2021). Therefore, any new research that improves or increases the ease of DDA balance could be a serious boon.

One underexplored challenge for DDA system tuning and evaluation is the assessment of outcome diversity, by which we mean how varied the outcomes of a system are in terms of player experience. DDA systems are typically designed to achieve a goal such as a target value for a difficulty heuristic. Even if a given DDA system is working perfectly it might not be clear how much outcome diversity is still present and what types of player experience could result. Even in the extreme case of having a fixed start and end point that a DDA system needs to guide a player between, many paths or action sequences are possible. Game designers naturally want to have knowledge and control over this, as too much or little diversity in the experiences delivered by the DDA system could be deleterious to the intended play experience.

Our work focuses on the challenge of evaluating the potential play experiences from DDA systems. We argue that

DDA research could benefit from utilizing methods from other research fields, especially from the related field of Procedural Content Generation (PCG)—a field also concerned with evaluating the game-play impact of possible outcomes from stochastic systems. Specifically, we explore the use of Expressive Range Analysis (ERA), a widely used PCG technique which allows for the visualization of output in terms of designer selected metrics (Smith and Whitehead 2010). It appears highly relevant to DDA system evaluation, although to our knowledge no prior work has explored this overlap, despite its popularity (Withington, Cook, and Tokarchuk 2024) and the wide range of content types it has found to be useful in assessing (Green et al. 2019; Guzdial et al. 2020; Kreminski et al. 2022).

In this work, we present a process that uses ERA as well as unsupervised clustering of play-traces to understand the outcome diversity of an in-development DDA system, *FighterDDA*. FighterDDA is a Turn-Based Role-Playing Game (TBRPG) being developed for both academic use and non-academic play. It has several DDA director modes that target different player audiences, and we apply our ERA and unsupervised clustering approach to compare and contrast these different modes in terms of the play experiences they offer. We demonstrate that applying this style of analysis is capable of producing nuanced and actionable design insights that improve core elements of the DDA system, allowing a designer to understand and address whether or not they are achieving their design goals.

Background

Dynamic Difficulty Adjustment

Dynamic Difficulty Adjustment (DDA) is an approach to dynamic game balancing that focuses on adjusting the difficulty of a game in response to the player and game state during runtime (Hunicke 2005). DDA is a heavily researched area of game balancing (Zohaib 2018), driven in part because of its ubiquity and diversity of methods in industry. An academic investigation of its use in industry at *EA Games* is documented by (Xue et al. 2017). DDA can take many forms depending on the design goals of the game. For instance, a DDA system in the game series *Mario Party* (Hudson Soft 1998) provides unexpected rewards to players based on gameplay metrics outside of ranking considerations, providing opportunities to include players of variable skills reasonably compete in a single game environment (Vicencio-Moreira, Mandryk, and Gutwin 2015). On the other end of the spectrum, DDA might seek to change the system to meet the player's current level of skill and/or gradually increase it over time to keep a player in a "flow" state (a state of challenge that keeps the player from being too bored from a lack of challenge and too frustrated from too great of a challenge) (Baumann, Lürig, and Engeser 2016; Larche and Dixon 2020). An example of this is in the game *Homeworld*(Sierra Studios 1999), where the volume of enemy spawns is increased if a player is successful in prior scenarios. A third design direction for DDA systems is the adherence to a desired dramatic pacing, such as in the game *Left 4 Dead*, where an AI director seeks to provide a moderation of perceived player tension such that there are alternating periods of increasing and decreasing difficulty, mimicking a dramatic curve (Booth 2009).

This last example is of note for this work as it employs the design pattern of "AI Directors" that act as experience management systems for the player (Satoi and Mizuno 2024). These systems utilize an often invisible agent known as a "director" that perceives a game state and uses parameter tuning approaches to achieve a goal given to the system (Thue and Bulitko 2018). A key attribute of dynamic game balancing overall (and as such its inheriting category of DDA and implementation of AI directors) is that it is dependent on designer goals (Schreiber and Romero 2021). This means that there is a diversity of approaches to any game balancing, DDA, or experience management system —using the examples above, a director might seek to include various skill levels, linearly ramp to meet increased player mastery, or modulate difficulty up and down to match some dramatic curve. The variety of designer goals and the procedural nature of these dynamic systems implies that such systems have a range of expressivity based on how they are implemented, and that expressivity is a critical aspect of evaluating the potential success of a given DDA system.

Expressive Range Analysis

Expressive Range Analysis (ERA) is a widely used technique in procedural content generation research for visualizing the possible output from systems for generating game content. It was introduced by Smith and Whitehead (Smith and Whitehead 2010) and its popularity is at least partly due to the simplicity of its operation. To apply ERA you need a sample of generated content and two or more quantitative metrics which can be calculated for each piece of generated content. These metrics values can then be used to position each piece of content on a 2D plot, most commonly a heat map. This allows a designer to visualize the output space of possible content in terms of metrics of interest; highlighting types of content which are likely, unlikely, or never seen as generated output.

When applying ERA to game content, metrics have historically been heuristics for difficulty or an aspect of player experience, taking direct inspiration from Smith and Whitehead's original work. However the original work makes clear that metrics should be domain specific and related to designer intention. ERA has been applied to diverse content types including video game dungeons (Green et al. 2019), NPC conversations (Morrison and Martens 2017) and authored poetry (Kreminski et al. 2022). While it has been applied to play-traces to explore aspects such as game balance (Withington and Tokarchuk 2023) and play style diversity (Guzdial et al. 2020) it has not to our knowledge been applied to play-traces from DDA systems.

Mixed-Initiative PCG

In this work we take heavy inspiration from the field of Mixed-Initiative Procedural Content Generation (MI-PCG). MI-PCG refers to systems which aim to support the co-creation of game content with participation from both a human designer and an autonomous or semi-autonomous sys-

tem (See (Lai, Fol Leymarie, and Latham 2022) for a recent survey of the whole field). MI-PCG research is concerned with both the design of such systems and how to integrate new algorithms, but also on game designers' experience and expectations of co-creation (Guzdial, Liao, and Riedl 2018; Lai, Latham, and Leymarie 2020). To our knowledge, no academic work has been done on integrating MI-PCG ideas with DDA research, and the intersection of the two is one we are interested in exploring in this and future work.

While we do not consider this system to be Mixed-Initiative as there is no agent present which can take independent action, we have similar goals in that we want to design a system that allows a designer to iteratively tune a DDA system to produce desirable encounters. As a result we do take inspiration from MI research, especially the concept of the iterative loop in which a piece of content or content generator is steadily improved or more closely aligned with designer goals through an iterative process of changes into system responses. (Lai, Fol Leymarie, and Latham 2022) considered this loop to be the most definitionally important trait of MI systems and it is one we make central to our system design. We place our work between MI and play-trace clustering methodologies, which typically use aggregate gameplay data over temporal progression in order to identify trends and patterns in player behaviors and game-state (Valls-Vargas, Ontanón, and Zhu 2015). Play-trace analysis has been used in context of DDA, such as in (Andersen, Gulwani, and Popovic 2013), where traces were used to manage educational progressions.

Drama and Tension in Game Systems

The control of tension in games is a well-researched area of game design that uses a paced attribute of a game (e.g. narrative, difficulty) to match some desired curve of the designers. The concept has frequently been used in the context of narrative games, where a (potentially procedural) game narrative meets requirements of storytelling concepts such as rising action or exposition (Riedl and Bulitko 2013). In terms of narrative, these examples highlight how dynamic systems seek to have certain properties of tension while enabling diversity in gameplay experiences. Connecting the dots between narrative and systematic tension, (Silva, Cardoso, and Oliveira 2019) highlights how these components work together (and sometimes are at odds with one another) to form a holistic game experience. In this vein, the intensity graph of games such as *Left 4 Dead* (South 2008) mimic the story curves of narrative as they are embedded in a game-mechanics and -systems perspective (Booth 2009). (Mejerwall 2025) makes this explicit in a talk on the implementation of AI directors, where the control of enemy spawns are set on a roughly sinusoidal pattern that is smoothed into statically-authored moments in order to give players a cyclical feeling of tension build-up and release. As such, AI directors represent an interesting case-study for the investigation of how a DDA system can both achieve a dramatic goal while retaining a sense of gameplay variance.

FighterDDA

FighterDDA is a simulation testbed that simulates a Turn-Based Role-Playing Game (TBRPG) fight between two teams, which facilitates either a player vs. player or player vs. AI agent battle. FighterDDA focuses on the TBRPG genre as it has a long history of success in industry in series such as *Final Fantasy* (Square 1987) and *Pokémon* (Game Freak 2022). The genre is interesting for ERA analysis as these games have a wide variety of expressivity depending on the style of combat encounter desired by the designer —easy farming encounter, mini-bosses and bosses, or player vs. player adversarial matches. FighterDDA is a headless system, enabling it to run thousands of simulations on the order of minutes. It emulates a common battle system used in games such as *Final Fantasy VII* (Square 1997), where an action meter gradually fills up for each character. When the meter is full, the character is allowed to perform an action from a list —attack, magic, defend, heal, etc. The battle continues until all characters from a single team have their health reduced to zero. Each team is initialized using a rubric of different character archetypes with unique stat and action potentials and fuzziness applied to initial stat creation.

System Design

The system simulation runs the game in units called "time steps". Each time step, character speeds are evaluated to see if they have reached a threshold where an action can be queued for execution. If multiple characters reach the threshold at the same time, ties are broken by how great the threshold was broken, and then by base character speed. If all of these are tied, action order is picked randomly. When a character action is ready to be queued, a utility agent is used to select what specific action should be performed based on the game state. Each team has an independent agent controlling it, and utility scoring is concerned with eliminating enemy characters while preserving ally health totals. Once an action is selected, it is then added to an execution queue, which applies the action at a specified future time step. Simultaneously and at a regular interval, a separate director agent is evaluating the current game state and deciding the magnitude and style of the balancing action to perform.

The director agent currently operates in three modes, i.e., **inclusion**, **difficulty-player**, and **difficulty-environment**. These modes are described in further detail in the section below. Player agents also have different operation modes which can be used to emulate players of different skill levels. For the purposes of our evaluation, we use either an "optimal" agent that represents an experienced player and a "random" agent that represents a novice player depending on the director condition. Time steps are repeated until all characters on one team have been defeated or the game reaches a maximum time (approximated to 30 minutes of human play-time). A diagram representation of the simulation along with the interaction with data visualization and designer input is shown in Fig. 2. A visual mockup of what the frontend of the system will look like is shown in Fig. 1. A more detailed description of the system's design, agent behavior, and initial testing can be found in (Shields and Melcer 2025).

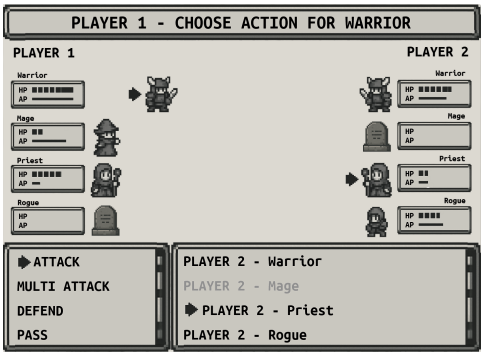

Figure 1: A mockup of the frontend of FighterDDA.

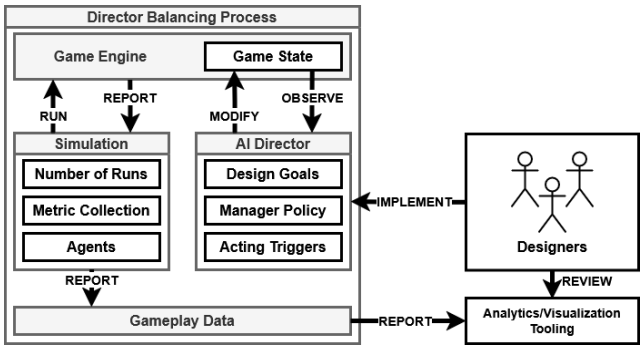

Figure 2: A diagram showing the core loop of the FighterDDA system.

System Goals

FighterDDA is meant to evaluate the effectiveness and game types enabled by the introduction of a variety of AI directors within the combat systems of TBRPGs. These directors have varied goals and methods to achieve these goals, aligning to earlier points in the related work section on DDA that a single game platform can contain a variety of dynamic game balancing approaches. The director follows the standard experience manager pattern —given a goal for a desired gameplay experience, an agent evaluates the current game state and then applies changes to the game world in order to meet its goals. In this paper, we evaluate the performance of three director styles in particular, with varied goals and strategies to achieve them:

- ***Inclusion*** – Aims to make games closer and therefore more dramatic for two players of differing skill levels. Achieves this through adjustment of character stats and application of targeted damage and heals. Uses one optimal and one random player agent.
- ***Difficulty-Player*** – Aims to make games more/less difficult for two players of similar skill levels. Achieves this through the adjustment of character stats. Uses two optimal player agents.
- ***Difficulty-Environment*** – Aims to make games more/-less difficult for two players of similar skill levels. Achieves this through the adjustment of environment attributes such as damage scaling. Uses two optimal player agents.

The designer of FighterDDA would be able to understand the character of fights produced by these director conditions, providing signals for what types of dramatic experiences they can provide for players based on which is active. The designer should be able to understand the variation within a given designer setting, ensuring that a given setting can still produce a diversity of experiences while still achieving some specific experience goal. In the following sections, we document an approach that investigates and categorizes the expressive character of each of these director settings in order for a designer to answer such questions about the range of possible play-traces in the FighterDDA system.

Methodology

Here we present our approach for analyzing and evaluating the play experiences delivered by a work in progress DDA system. The intention is to give game and systems designers the information they need to confirm whether or not a DDA system is going to give players the intended experience or set of experiences using visualization and analysis techniques beyond raw stat reporting. Therefore, the priorities for designing this workflow were first to privilege designer intention and allow for the fact that there are innumerable goals that a designer could have for a given system, and second to deliver information about the system in a way that can be easily understood in terms of these goals.

Our approach involves three primary steps which are arranged in linear series. Although this series is iterative and can be re-run at any time if a design flaw or idea is noticed and corrected. We feel this process best reflects the iterative tuning and design which is typical in real world game development, and also accommodates the difficulty of tuning DDA systems which have to support diverse players and play experiences through the balancing of many parameters and mechanics. A high-level overview of this process is available in Fig. 3.

While the DDA system itself is built in JavaScript, the evaluation system is developed in Python. The full platform and all tooling presented in this work are available on an online repository hosted on GitHub [1].

Play-trace Collection

To facilitate analysis, we first need to gather play-traces. FighterDDA was designed to deliver players combat encounters with autonomously controlled opposing teams which possess access to the same combat capacity and abilities as the player. To gather play-traces from the system, we tasked two autonomously controlled teams with the efficient defeat of the opposition and then recorded the results. Any time a character in either team took an action, this action and the associated game state were logged to a JSON file, enabling in-depth analysis of how the fight progressed.

We evaluated the three different modes of operation for the DDA system (See Section FighterDDA), as well as a

[1] https://github.com/smshields/FighterDDA - ERA and graphing tools are under the ”tools” directory

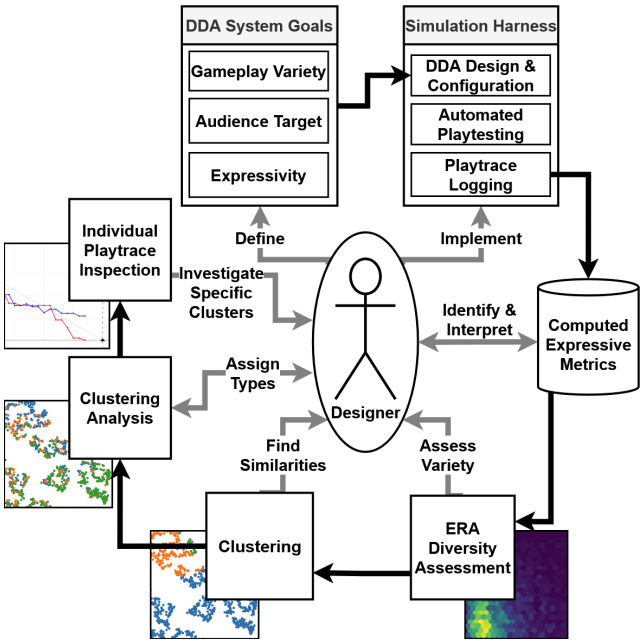

Figure 3: An overview of the process of analyzing the expressivity of a DDA using a simulation testbed and ERA methods. The designer is at the center of the process and can both define and derive insights from each step —they decide and evaluate metrics, get insights on gameplay variety and typing from ERA and clustering exercises, and can investigate specific play-traces.

'No-Director' mode to explore the functioning of the system when all DDA adjustments were deactivated. For each mode, we generated 1,000 play-traces to use in the system evaluation as it was felt that this struck the right balance between exploration and speed of generation.

Metric Calculation

For the experiments presented here we calculate five metrics for each play-trace, each of which is explained in Table 1. As ERA has not been applied to play-traces previously there is no precedent for what metrics should be used. We instead relied on the intuition of the system designer and exploratory analysis of what appeared to give interesting clusters.

The first two metrics (Total Time Steps and Absolute Stat Difference) are focused on system functioning and are used for conducting ERA. Whereas the second set of metrics (Remaining HP, Damage per Step, and Lead Changes per Step) are calculated to try and capture the dramatic character of encounters and are used in unsupervised clustering.

Step 1: ERA Diversity Assessment

The first step of the analysis is to confirm whether FighterDDA can produce the desired diversity of play experiences in terms of encounter length and initial combat ability. To do this we use Expressive Range Analysis (ERA) to visualize all play-traces in terms of Total Time Steps and Absolute Stat Difference. These were calculated to explore the distribution and bounds of each, as we want to ensure that encounters are neither too long nor too short, and that teams are initially balanced enough while not being identical. We were also interested in the relationship between the two metrics. For example, a negative linear correlation between the two could indicate the DDA system was failing to compensate for the ability difference between teams.

Metric Name	Description
Total Time Steps	The length of the encounter.
Absolute Stat Difference	The difference in performance stats between teams at beginning of game.
Lead Changes Per Step	Average amount of times the team with the highest percentage of total health remaining per time step.
Damage Per Step	Average amount of damage done by both teams combined per time step.
Winning Player Remaining HP	Remaining health of the winning team as a percentage.

Table 1: A list of metrics generated by game simulations and used for ERA and clustering approaches.

Step 2: Encounter Clustering

The second step used in our analysis is the unsupervised clustering of metrics related to the dramatic elements of play-traces, specifically of the three drama-linked metrics we highlighted in the above section on metric calculation. The goal of this step is to describe the types of dramatic encounters supported by FighterDDA. Whereas in Step 1 we were interested in the distribution and bounds of the metrics here we are using them to discretize the play-traces into distinctive sets which contain a shared dramatic experience. The inspiration for this approach came from the writing of people such as (Brazie 2024) on the types of enemy encounters, as well as the work on drama curves.

While many unsupervised clustering algorithms exist, we opted to use K-Means clustering, specifically scikit-learn's implementation [2], as in our initial testing it seemed to perform better than the alternatives such as DBSCAN on our data in terms of cluster robustness, but also because it is easily interpretable, with each point belonging to a single cluster. The only parameter in K-Means is the number of clusters (k), which we determined using the 'elbow method', i.e., Calculating the inertia for each possible k value and looking for the plateau point.

We then aimed to make these clusters interpretable, which we do by producing summary statistics for each cluster in terms of the drama metrics used to produce them. We calculate the mean and standard deviation for each metric for each cluster, as well as the minimum and maximum values. This gives us the information they need to understand what

[2]https://github.com/scikit-learn/scikit-learn/blob/da08f3d99/sklearn/cluster/_kmeans.py

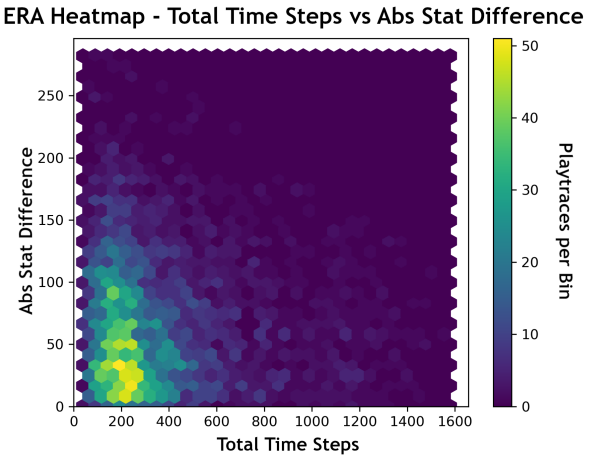

Figure 4: ERA Heatmap of the relationship between Total Time Steps and Absolute Stat Difference.

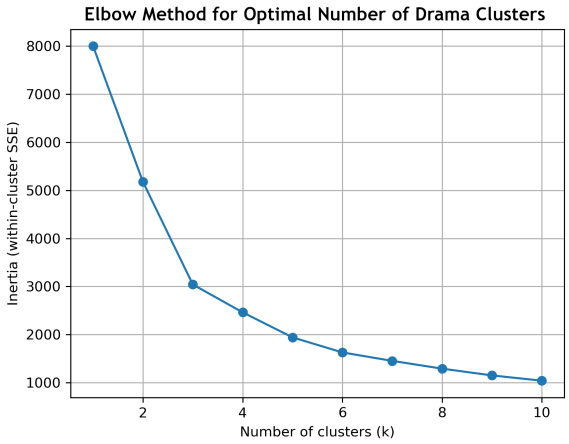

Figure 5: Visualization of the inertia of various K-Values when applied to the drama metric dataset.

types of play-trace are found in each cluster, but the interpretation process is manual, a limitation we discuss further in the Limitations & Future Work.

Step 3: Cluster Analysis

The goal of our final analysis step is exploring whether the encounter types discovered in Step 2 also contain the diversity which was identified in Step 1, as well as exploring whether all clusters are possible from all DDA modes. Without this step we could miss that while the FighterDDA system is delivering a desirable set of encounter types with the correct level of overall diversity, that some of these encounter types present players with highly predictable experiences.

To assess this, we reconduct the ERA visualizations produced in Step 1, except that we differentiate them by the cluster membership produced in Step 2. In case the visualization is unclear, we also produce the same summary statistics as in Step 2 except for Step 1's metrics. Notably, Step 3 involves effectively repeating Step 1's analysis. However, we still argue that Step 1 is valuable and appropriate as a primary step in this method as it is comparatively quick to conduct and finding a negative result would negate the need for Steps 2 or 3.

Results

In this section, we describe the results we attained from applying our methodology to the FighterDDA system. We do this to both demonstrate the strengths and limitations of the approach when it is used on a real in development game system, and to give a more tangible idea of how similar approaches could be applied in alternative domains.

Step 1 Results

The ERA heatmap plot in Fig. 4 clearly shows a strong tendency within FighterDDA encounters, with a strong cohesive hotspot in the bottom left of the plot roughly between 50 <Total Time Steps <250 and 0 <Absolute Stat Difference <70. It also shows the extreme variance in the outlying portions of the expressive space, with a handful of encounters being much longer or with very mismatched starting stats.

Cluster	Metric	Mean	Stddev
0	Winning Player HP	**173**	36.8
	Lead Changes Per Step	0.017	0.0101
	Damage Per Step	2.49	0.901
1	Winning Player HP	117	51.1
	Lead Changes Per Step	**0.0498**	0.0221
	Damage Per Step	**7.77**	2.79
2	Winning Player HP	69.5	36.3
	Lead Changes Per Step	0.0273	0.0134
	Damage Per Step	1.85	0.759

Table 2: The mean and standard deviation of the metric values for each of the three clusters found through applying K-Means to the drama metrics. The highest mean values are highlighted in bold.

Step 2 Results

The K-Means elbow visualization in Fig. 5 appears to show plateauing at K=3 so this is the number of clusters we opted for. We should note however that it is not a very strong elbow which could indicate weaker cluster membership. To assess this further we also calculated the silhouette score of K=3 which gives a score from -1 to 1 on how strongly differentiated the clusters are. We calculated a value of 0.439 which confirms that the cluster membership is relatively weak. That being said, the clustering produced readily mapped on to gameplay styles and corresponded well with the director

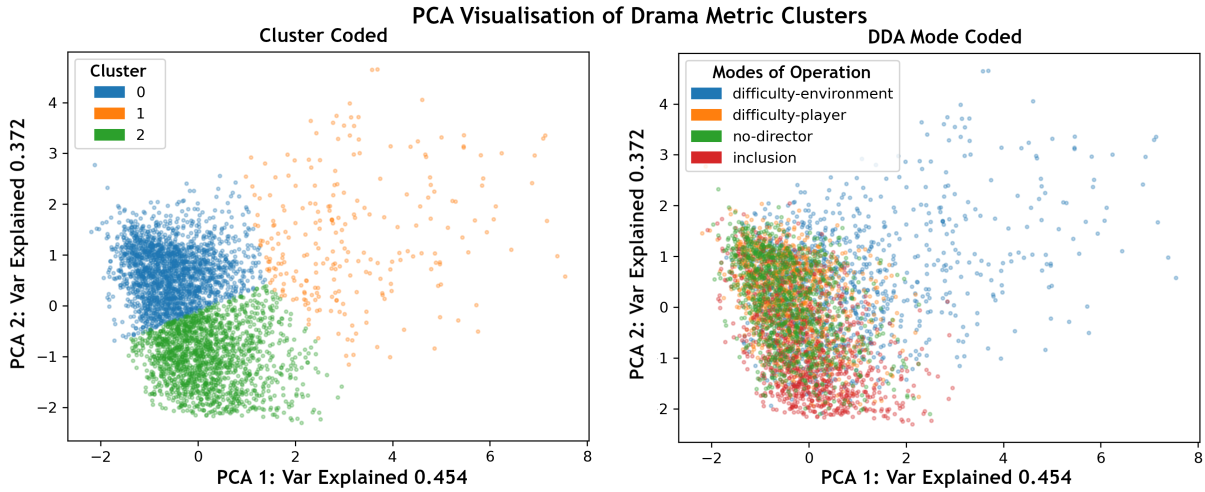

Figure 6: Visualization of the first two Principal Components derived from applying PCA to the set of drama metrics. Color-coded according to cluster (left) and DDA mode source (right).

conditions specified (see Discussion), indicating that even with a relatively weak clustering it still produced valuable insight into the system performance.

We applied K-Means clustering to the drama metric set with k=3 to categorize all play-traces, and then visualized these clusters after applying PCA to the metric set to reduce the set to two dimensions from three (Fig. 6, which was done while still explaining 82.6% of the variance in the underlying set. This visualization shows that there are two dense and well formed clusters in the space, labeled 0 and 2, and one diffuse cluster with lower play-trace membership labeled 1. We also contrast this in the same figure with the same points color-coded based on the DDA mode that produced them, which shows significant overlap in all modes in their cluster 0 and 2 membership, and that cluster 1 is almost exclusively produced by mode 'difficulty-environment'.

We also produced the summary statistics for each cluster displayed in Table 2. Our interpretation of the metrics gives us the following characterizations of the dramatic clusters supported:

- **0: *Grindy***: High HP remaining, low lead changes, low damage per step. Boring, low-tension, drawn-out. Comparable to a battle with a weak enemy team that has a large health pool.
- **1: *Explosive*:** Medium HP remaining, very high lead changes, high damage per step. Dramatic, tense, swingy. Comparable to a battle with two teams of equal strength and high potential to punish mistakes and leverage advantages.
- **2: *Strategic*:** Low HP remaining, medium lead changes, low damage per step. Endurance-oriented, neck-and-neck, tactical. Comparable to a traditional "boss" encounter —long games with smaller HP ratio changes over time that have a sense of tension and strategy.

Step 3 Results

Finally, for Step 3 we return to Step 1's expressive range analysis but this time in scatterplot form, allowing us to color code points based on both DDA mode and K-Means cluster (Fig. 7). This shows that while the overall set is highly diverse in terms of these two metrics, this diversity does not translate to all clusters. Cluster 1 appears to be highly consistent in the length of encounters, being heavily clustered between 50 and 100 time steps, as well as more clustered than is typical in terms of absolute difference. The fact that this is a peculiarity of a single DDA mode is clarified by the DDA mode coding of this set, which shows that the 'difficulty-environment' mode is producing encounters which are both dramatically distinct as highlighted in Step 2's results, but also mechanically distinct in the time they take to conclude, as indicated by their heavy clustering to the left of the mode coded plot.

Discussion

Rapid System Improvement and Understanding

The iterative loop proved a boon to the design of Fighter-DDA throughout the entire process. The production of Step 1's ERA heatmap (Fig. 4) helped confirm that the system had a relatively predictable game length depending on the gap in stats noted between teams while retaining the potential for "underdog" stories where one team with a higher stat difference lasts unexpectedly long or takes the win. Figures 6 and 7 show clear relationships between different director conditions and the styles of games produced, allowing for easy observations of how a given director condition produced a given range of play experiences. The spread of such plots also gives a quick understanding of the consistency and variety of output for a condition —while cluster 1 (explosive) games are desirable by the designer, they are far more diffuse in terms of dramatic measurements (Table 2) and mostly constrained to the difficulty-environment con-

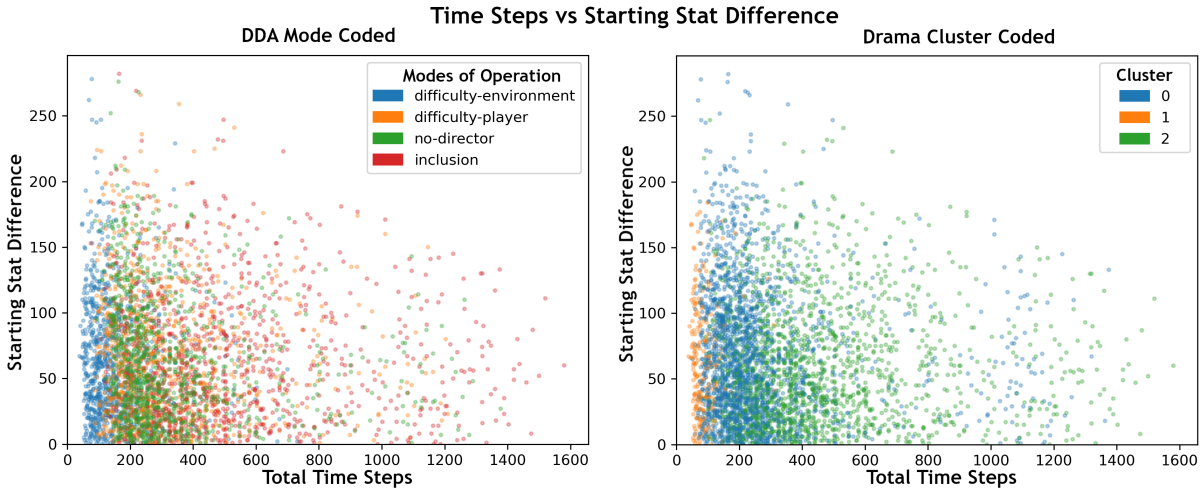

Figure 7: ERA Scatterplot color-coded by DDA mode of operation (left) and drama cluster (right)

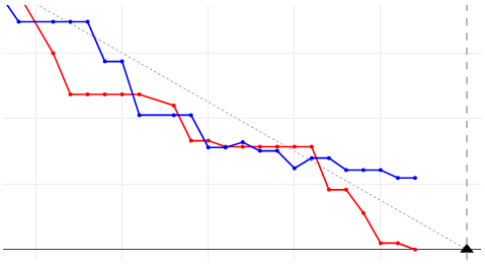

Figure 8: FighterDDA playtrace graphing tool to inspect details of specific games.

dition. This tells the designer that this condition is doing its job to create a specific play-style, but may also create a wide variety of potential play-traces.

The plotting also allowed the designer to evaluate how well the difficulty conditions performed their job. For data collection, the director aimed to produce games of low-length between players of similar skill. In the difficulty-environment condition, this goal was achieved and achieved consistently, as shown by the vertical blue grouping of points in the DDA Mode Coded graph in Fig. 7. However, the difficulty-player condition seemed to fare much worse, providing a much more diffuse spread of game lengths that was potentially even less well distributed than the no-director condition. This revealed to the designer that the difficulty-player condition was not performing its functionality as intended, even if surface-level metric collection had initially obfuscated this fact for the designer.

Understanding Drama Through Automated Playtesting and ERA

Beyond testing design hypotheses and determining successful director strategies, our methodology allowed for investigation of system drama in a way that is normally only readily apparent during human playtesting. Our three clusters (grindy, explosive, strategic) are easily mapped onto various conventions in TBRPG design and represent key ways encounters can be shaped to impact overall player tension and progression during a game (Stenström and Björk 2013). Grindy games might initially be viewed as a net negative, but they are used as a pattern in games where player progression is gated by the acquisition of experience points or enemy item drops (Sgandurra 2022). Explosive games produce tense environments where simple mistakes can lead to major changes in game state, making every decision critical and meaningful during play. This is valuable in competitive contexts in particular (such as the *Pokémon* (Game Freak 2022) series' Video Game Championship competitive format (Burch and Lee 2024), where games have diverse playstyles that are both somewhat stochastic and sensitive to mistakes (Angliss et al. 2025). Strategic fights land in the archetypal "boss" category, where a longer game with subtler changes in damage and winning positions leads to a period of protracted intensity and a close finish between teams —one player *barely* survives the encounter (Agriogianis 2018).

The insight that the difficulty-environment director leads to fights of the explosive (type 1) nature is valuable and means that it would be an effective approach in competitive environments. This fulfilled the designer's goals while providing additional positive findings —similarly skilled players (optimal agents) should have games in a relatively regular time window with lots of drama (as represented by damage dealt and changes in leading characters). Meanwhile, the lack of differences between the no-director and difficulty-player conditions indicates that this condition is ineffective in both its functional output as well as its generative potential.

Finally, the inclusion condition produced particularly meaningful results —not only did it have differentiation from the difficulty- and no-director conditions in how game length was distributed, but it appeared to have a unique dramatic profile from them as well, landing squarely in the

strategic (type 2) bucket. This is reassuring for the goals of the condition, which seeks to make players of unequal skill levels compete on an even playing field. As noted earlier, the inclusion condition features player agents of both the optimal and random variety, meaning that without the director involved the optimal player would consistently win over the random player. As the play-traces show longer games with a small difference in remaining HP, two things become clear: First, the director is effectively closing the skill gap between the two player types. Second, the increased game length indicates that making individual mistakes would be less prone to extreme punishment as demonstrated in the type 1 clustering, creating a more forgiving environment for players. In a player vs. AI encounter, this director would also meet goals of fulfilling the "boss" style of encounter by making the player feel as if they are always in danger of losing or winning against an opponent. (Gonçalves et al. 2024) note that this quality of balancing such that players are constantly just-behind or just-ahead increases engagement and enjoyment of experiences overall.

Macro- and Micro- Visualizations in Procedural Game Systems

Understanding a procedural game system is a difficult task —while the methods to generate a game element or system might be clear, the full range of possibilities of those methods might be incomprehensibly large. Applying visualization and data-processing techniques to procedural systems helps negate the chaotic nature of testing these systems while removing the risk that a designer is exposed to misleading outliers. Being able to, at a glance, understand both the variety and dramatic character of a procedural balancing system through a visualization of a high volume of play-traces means that designers spend less time attempting to interpret random sets of generation and instead focus on targeted changes that will meaningfully impact overall experience of their system.

A final benefit of our approach is that as a requirement to generate the macro-visualization of all play-traces, each individual play-trace is comprehensively documented, annotated, and ready to be plotted itself. Such a plot is shown in Fig. 8, which allows the investigation of any given point on the ERA plots by looking at how leads changed, how frequently and to what extent did a director act, and so on. In this way, there is a clear pipeline to our process —after ERA and cluster analysis is complete, designers can inspect particularly interesting outliers or pick samples from clusters to get further clarification on what games of a certain type look like. This combination of macro- and micro- views of automated playtesting provides a wide range of co-creative tools to tune and optimize the exceedingly complex area of dynamic game balancing systems.

Limitations & Future Work

A limitation of our approach is our reliance on automated playtesting to gather play-traces and the unknown relationship they have with those one would get from human survey. Automated playtesting is highly defensible from a resourcing perspective, as once the development has been done to support it it is easy to scale and costs no time or money to retest unlike human playtesting. However, it would be valuable to know how much of an abstraction this is, and in future work we intend to conduct studies using human players to investigate the extent to which they explore similar encounter possibility spaces to automated agents.

An opportunity to confirm our tooling approach's contribution to the MI-PCG space would be to conduct user studies where game designers are actively attempting to implement some form of DDA system and have play-traces available. In-depth case studies of our methods in more varied environments would help more concretely show that ERA can prove effective in systems-oriented contexts.

A practical limitation of our approach is the reliance on human interpretation of the play-trace clusters produced in Step 2 of our method. As we were only clustering on three metrics and only three clusters were produced, there were not too many data points to interpret. However there is no upper limit on the amount of metrics that one could hypothetically use in this process, and any increase would complicate interpretation.

Related to the above limitation, is the requirement for human designers to design the metrics of interest. This places a configuration burden on designers and also limits the evaluation to what can be conceived of and actually calculated. This limitation is harder to overcome, though we do have options to explore, including ERA selection criteria (Withington and Tokarchuk 2023) to decide on metric pairings for conducting ERA, as well as sourcing metrics from validated sets (Mariño, Reis, and Lelis 2015).

Conclusion

In this work, we documented an approach to investigate the expressivity of a procedural system as it applies to dynamic game balancing. We applied a multi-step ERA and clustering approach to a varied set of AI director implementations to understand their dramatic qualities and ability to fulfill key designer goals. We found that the approach was successful in providing detailed and surprising insights into what was and was not working for each director condition, and provided concrete direction for future tuning approaches. We argue that our approach holds potential to be useful in tuning dynamic game balancing systems that prioritize specific expressive traits such as tension or drama.

Acknowledgments

This work was supported by the EPSRC Centre for Doctoral Training in Intelligent Games & Games Intelligence (IGGI) [EP/S022325/1].

This material is also based upon work supported by the National Science Foundation under Grant No. 2202521. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

Agriogianis, T. 2018. The Roles, Mechanics, and Evolution of Boss Battles in Video Games.

Andersen, E.; Gulwani, S.; and Popovic, Z. 2013. A trace-based framework for analyzing and synthesizing educational progressions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 773–782.

Andrade, G.; Ramalho, G.; Gomes, A.; and Corruble, V. 2006. Dynamic game balancing: An evaluation of user satisfaction. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 2, 3–8.

Angliss, C.; Cui, J.; Hu, J.; Rahman, A.; and Stone, P. 2025. A Benchmark for Generalizing Across Diverse Team Strategies in Competitive Pokémon. arXiv:2506.10326.

Baldwin, A.; Johnson, D.; Wyeth, P.; and Sweetser, P. 2013. A framework of dynamic difficulty adjustment in competitive multiplayer video games. In *2013 IEEE international games innovation conference (IGIC)*, 16–19. IEEE.

Baumann, N.; Lürig, C.; and Engeser, S. 2016. Flow and enjoyment beyond skill-demand balance: The role of game pacing curves and personality. *Motivation and Emotion*, 40: 507–519.

Booth, M. 2009. The ai systems of left 4 dead. In *Artificial Intelligence and Interactive Digital Entertainment Conference at Stanford, 2009*.

Brazie, A. 2024. Enemy design in games: A beginner's guide.

Burch, H.; and Lee, N. 2024. *Pokémon and World Championships*, 1431–1435. Cham: Springer International Publishing. ISBN 978-3-031-23161-2.

Dziedzic, D.; and Włodarczyk, W. 2018. Approaches to measuring the difficulty of games in dynamic difficulty adjustment systems. *International Journal of Human–Computer Interaction*, 34(8): 707–715.

EAD, N. 2014. Mario Kart 8. [DIGITAL].

Game Freak. 2022. Pokemon Scarlet and Violet. [DIGITAL].

Gonçalves, D.; Barros, D.; Pais, P.; Guerreiro, J.; Guerreiro, T.; and Rodrigues, A. 2024. The Trick is to Stay Behind?: Defining and Exploring the Design Space of Player Balancing Mechanics. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 1–16.

Green, M. C.; Khalifa, A.; Alsoughayer, A.; Surana, D.; Liapis, A.; and Togelius, J. 2019. Two-step constructive approaches for dungeon generation. In *Proceedings of the 14th International Conference on the Foundations of Digital Games*, 1–7. San Luis Obispo California USA: ACM. ISBN 978-1-4503-7217-6.

Guzdial, M.; Acharya, D.; Kreminski, M.; Cook, M.; Eladhari, M.; Liapis, A.; and Sullivan, A. 2020. Tabletop Role-playing Games as Procedural Content Generators. In *International Conference on the Foundations of Digital Games*, FDG '20. New York, NY, USA: Association for Computing Machinery. ISBN 978-1-4503-8807-8. Event-place: Bugibba, Malta.

Guzdial, M.; Liao, N.; and Riedl, M. 2018. Co-Creative Level Design via Machine Learning. Publisher: arXiv Version Number: 1.

Hudson Soft. 1998. Mario Party. [Nintendo 64].

Hunicke, R. 2005. The case for dynamic difficulty adjustment in games. In *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology*, 429–433.

Jaffe, A.; Miller, A.; Andersen, E.; Liu, Y.-E.; Karlin, A.; and Popovic, Z. 2012. Evaluating competitive game balance with restricted play. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 8, 26–31.

Jeon, H.-C.; Baek, I.-C.; Bae, C.-m.; Park, T.; You, W.; Ha, T.; Jung, H.; Noh, J.; Oh, S.; and Kim, K.-J. 2023. RaidEnv: Exploring new challenges in automated content balancing for boss raid games. *IEEE Transactions on Games*.

Kavanagh, W.; Miller, A.; Norman, G.; and Andrei, O. 2019. Balancing turn-based games with chained strategy generation. *IEEE Transactions on Games*, 13(2): 113–122.

Kokkinakis, A.; York, P.; Patra, M. S.; Robertson, J.; Kirman, B.; Coates, A.; Chitayat, A. P. P.; Demediuk, S.; Drachen, A.; Hook, J.; et al. 2021. Metagaming and metagames in Esports. *International Journal of Esports*, 1(1).

Kreminski, M.; Karth, I.; Mateas, M.; and Wardrip-Fruin, N. 2022. Evaluating Mixed-Initiative Creative Interfaces via Expressive Range Coverage Analysis. In *3rd Workshop on Human-AI Co-Creation with Generative Models*, volume 3124.

Lai, G.; Fol Leymarie, F.; and Latham, W. 2022. On Mixed-Initiative Content Creation for Video Games. *IEEE Transactions on Games*, 1–1.

Lai, G.; Latham, W.; and Leymarie, F. F. 2020. Towards Friendly Mixed Initiative Procedural Content Generation: Three Pillars of Industry. Version Number: 1.

Larche, C. J.; and Dixon, M. J. 2020. The relationship between the skill-challenge balance, game expertise, flow and the urge to keep playing complex mobile games. *Journal of behavioral addictions*, 9(3): 606–616.

Mariño, J.; Reis, W.; and Lelis, L. 2015. An empirical evaluation of evaluation metrics of procedurally generated mario levels.

Mejerwall, M. 2025. Growing an AI Director into a Full Adventure Director. Game Developer Conference.

Morrison, H.; and Martens, C. 2017. A generative model of group conversation. In *Proceedings of the 12th International Conference on the Foundations of Digital Games*, 1–7. Hyannis Massachusetts: ACM. ISBN 978-1-4503-5319-9.

Mortazavi, F.; Moradi, H.; and Vahabie, A.-H. 2024. Dynamic difficulty adjustment approaches in video games: a systematic literature review. *Multimedia Tools and Applications*, 83(35): 83227–83274.

Nintendo EPD. 2017. The Legend of Zelda: Breath of the Wild. [Nintendo Switch, Wii U, Nintendo Switch 2].

Reis, S.; Reis, L. P.; and Lau, N. 2021. Game adaptation by using reinforcement learning over meta games. *Group Decision and Negotiation*, 30(2): 321–340.

Riedl, M. O.; and Bulitko, V. 2013. Interactive narrative: An intelligent systems approach. *Ai Magazine*, 34(1): 67–67.

Satoi, D.; and Mizuno, Y. 2024. Meta artificial intelligence and artificial intelligence director. In *Encyclopedia of Computer Graphics and Games*, 1119–1126. Springer.

Schreiber, I.; and Romero, B. 2021. *Game balance*. CRC Press.

Sgandurra, S. A. 2022. Fight. Heal. Repeat: A Look at Rhetorical Devices in Grinding Game Mechanics. *Simulation & Gaming*, 53(4): 388–399.

Shields, S.; and Melcer, E. F. 2025. FighterDDA: A Simulation Testbed for Evaluating Director-Based Dynamic Balancing. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*. AAAI Press. In press.

Sierra Studios. 1999. Homeworld. [Windows, OS X].

Silva, I.; Cardoso, P.; and Oliveira, E. 2019. Narrative and gameplay: the balanced and imbalanced relationship between dramatic tension and gameplay tension. In *Proceedings of the 9th International Conference on Digital and Interactive Arts*, 1–8.

Silva, M. P.; do Nascimento Silva, V.; and Chaimowicz, L. 2017. Dynamic difficulty adjustment on MOBA games. *Entertainment Computing*, 18: 103–123.

Smith, G.; and Whitehead, J. 2010. Analyzing the expressive range of a level generator. In *Proceedings of the 2010 Workshop on Procedural Content Generation in Games - PCGames '10*, 1–7. Monterey, California: ACM Press. ISBN 978-1-4503-0023-0.

South, V. 2008. Left 4 Dead. [Windows, Xbox 360, macOS].

Square. 1987. Final Fantasy. [Nintendo Entertainment System].

Square. 1997. Final Fantasy VII. [PlayStation].

Stenström, C. D.; and Björk, S. 2013. Understanding computer role-playing games: A genre analysis based on gameplay features in combat systems. In *Second Workshop on Design Patterns in Games (FDG 2013)*.

Thue, D.; and Bulitko, V. 2018. Toward a unified understanding of experience management. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 14, 130–136.

Valls-Vargas, J.; Ontanón, S.; and Zhu, J. 2015. Exploring player trace segmentation for dynamic play style prediction. In *Proceedings of the AAAI conference on artificial intelligence and interactive digital entertainment*, volume 11, 93–99.

Vicencio-Moreira, R.; Mandryk, R. L.; and Gutwin, C. 2015. Now you can compete with anyone: Balancing players of different skill levels in a first-person shooter game. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 2255–2264.

Withington, O.; Cook, M.; and Tokarchuk, L. 2024. On the Evaluation of Procedural Level Generation Systems. In *Proceedings of the 19th International Conference on the Foundations of Digital Games*, 1–10. Worcester MA USA: ACM. ISBN 9798400709555.

Withington, O.; and Tokarchuk, L. 2023. The Right Variety: Improving Expressive Range Analysis with Metric Selection Methods. In *Proceedings of the 18th International Conference on the Foundations of Digital Games*, 1–11. Lisbon Portugal: ACM. ISBN 978-1-4503-9855-8.

Xue, S.; Wu, M.; Kolen, J.; Aghdaie, N.; and Zaman, K. A. 2017. Dynamic difficulty adjustment for maximized engagement in digital games. In *Proceedings of the 26th international conference on world wide web companion*, 465–471.

Zohaib, M. 2018. Dynamic difficulty adjustment (DDA) in computer games: A review. *Advances in Human-Computer Interaction*, 2018(1): 5681652.