Adaptive Generative Difficulty Algorithm in Games

Khoi Tran

April 2025

A. Abstract

Dynamic video game challenge level adjustment plays a central role in player interaction, retention, and enjoyment while playing a video game. This project introduces the Adaptive Generative Difficulty Algorithm (AGDA), a novel framework designed to dynamically adjust game difficulty based on individual player performance and gameplay tendencies. In a modular Python implementation, the AGDA continuously monitors player stats such as hit rate, damage received, resource consumption, and exploration patterns to modify game parameters such as monster attributes and spawn rates. In a feedback loop, the algorithm balances between difficulty and accessibility to accommodate diverse playstyles such as aggressive, defensive, and balanced, among others. It also accounts for performance oscillation and streak-based modifiers in order to produce smooth transitions among difficulty levels. Nine-scale simulations across a wide range of playstyles, ranging from overachiever to underperformer, highlight the versatility and robustness of the algorithm, with the results shown in detailed graphs. The findings demonstrate that the AGDA effectively keeps players interested through adaptation of difficulty, with implications in procedural content generation and adaptive game design. This study contributes to game development research by presenting a scalable, data-driven difficulty scaling method that is player experience centered.

B. Introduction

Game design strives for a balance between challenge and fun that aligns with Csikszentmihalyi's theory of flow, where maximum engagement occurs when the challenge of an activity matches an individual's skill level (Csikszentmihalyi, 1990). Achieving this balance is challenging due to the diversity in player abilities, preferences, and behaviors. Traditional game difficulty mechanisms, often limited to fixed settings like "easy," "normal," or "hard," or requiring manual adjustments, frequently fail to accommodate this diversity, resulting in experiences that are either too difficult or too easy (Adams, 2010). Dynamic Difficulty Adjustment (DDA) systems, as seen in games like *Left 4 Dead* (Valve, 2008) and *Resident Evil 4* (Capcom, 2005), address this by modifying game parameters in real-time based on player performance (Hunicke & Chapman, 2004). However, these systems often rely on simple metrics, such as health or task completion time, lacking the sophistication to adapt to nuanced behavioral patterns or ensure smooth difficulty transitions, which can disrupt immersion.

This research introduces the Adaptive Generative Difficulty Algorithm (AGDA), a novel DDA model designed to deliver a personalized and seamless gaming experience in the case study game, *Legends of Đại Việt: The Shattered Lotus.* Set in a fantasy ancient Vietnam, this RPG focuses on exploring Ngọc Sơn Island, a temple on a foggy lake guarded by monsters derived from the game’s desert biome enemies. The AGDA adjusts game elements dynamically using a broad set of player metrics, including attack accuracy, damage received, resource utilization, and exploration progress. Unlike traditional DDA systems that apply binary easy-hard adjustments, AGDA accounts for behavioral tendencies: aggressive, defensive, or balanced playstyles, and incorporates

performance trends, volatility, and streak-based modifiers to fine-tune difficulty with precision (Drachen, Canossa, & Yannakakis, 2009). By employing a momentum-based feedback system and advanced trend analysis, the algorithm makes sure gradual, contextually appropriate changes to opponent attributes (health, damage, speed, spawn frequency) and player abilities (attack range, defense). This approach aims to maintain a consistent state of flow, enhancing player satisfaction and engagement across diverse skill levels and playstyles in the immersive world of Ngọc Sơn Island.

The AGDA's design is rooted in player-centric game design principles, aiming to create a tailored experience that resonates with the cultural and mythological themes of Legends of Đại Việt: The Shattered Lotus. Building on prior development, monsters were established as formidable adversaries in the game's desert levels, with their attributes formalized for dynamic scaling. The algorithm was tested through simulations across nine playstyles: aggressive, defensive, balanced, underperforming, overachiever, reckless, tactician, erratic, and stagnant using a robust analytical model with graphical visualizations to evaluate its responsiveness and effectiveness. This research contributes to game development by offering a scalable, empirical approach to adaptive difficulty scaling through player-centered design. Beyond video games, AGDA has potential applications in interactive systems requiring personalized challenge modulation, such as educational gaming and procedural content generation (Togelius, Yannakakis, Stanley, & Browne, 2011). The problem statement, evaluation plan, and feature selection process are addressed in subsequent sections to analyze the algorithm's effectiveness and key adaptability metrics.

C. Problem Statement

The primary challenge of video game design is the development of an adaptive difficulty mechanism that responds to players' variable and changing capabilities, preferences, and behavioral tendencies in a way that offers an exciting and engaging experience without annoying or boring them. Most current implementations of DDA, while novel, often falter on two key dimensions. First, there is a tendency to focus either on performance measures (e.g., successful attacks and damage received) or behavioral cues (such as aggressive actions) and not integrate both within a cohesive adaptive framework. This limitation narrows the scope to provide personally tailored gaming experience. Second, many DDA implementations face difficulty in achieving smooth difficulty progression, resulting in abrupt changes that detract from the player's immersive experience and also interfere with the achievement of flow, as expressed by Csikszentmihalyi (1990).

In the context of *Legends of Đại Việt: The Shattered Lotus*, this challenge is set up on Ngọc Sơn Island, where players confront monsters and navigate a continuously changing environment that is marked by both combat and exploration. The need for a system that is capable of dynamically altering the parameters of such enemies (i.e., speed and frequency of attack) and environmental obstacles (i.e., rates of enemy spawning) in response to real-time player activity and inferred behavioral patterns is required to ensure narrative fidelity as well as enhance player engagement. The central research question of this research is: How can game difficulty be smoothly and finely adjusted to fit a player's

evolving skills and behavioral patterns in a culturally enhanced, platformer game setting? The AGDA seeks to remedy this by combining performance metrics (attack success, damage taken, resource use, exploration) with behavioral inferences (aggressive or defensive inclinations) to dynamically adjust several aspects of the game to provide a tailored and integrated experience that maximizes satisfaction and immersion in the mythological realm of Đại Việt.

D. Proposed Evaluation

In order to thoroughly examine the effectiveness of the Adaptive Generative Difficulty Algorithm (AGDA), an extended evaluation plan is suggested, comprising quantitative simulations, qualitative analysis, and comparative benchmarking. Evaluation is engineered to confirm the algorithm's capabilities for maintaining player engagement, offering unobtrusive transitions in difficulty, and adapting to varied behavioral patterns, while also offering indications of its scalability and use in the context of Legends of Đại Việt: The Shattered Lotus.

Simulation-Based Evaluation: The primary evaluation method is comprehensive simulations within a structured analytical setting. The AGDA will be assessed according to nine various behavioral patterns: aggressive, defensive, balanced, underachiever, overachiever, reckless, tactician, erratic, and stagnant via 150 simulation cycles each, simulating player interactions with monsters at Ngọc Sơn Island. Key metrics, including difficulty level, adversary attributes (vitality, attack strength, speed, attack frequency,

recovery rate, attack range, recognition range), maximum possible adversary spawn rates, and player performance metrics (attack accuracy, damage received, resources used, exploration progress), will be quantified and graphed. Smoothness of difficulty transitions will be quantified by analyzing the variance and maximum cycle-wise changes in difficulty level, with a focus on changes within ±0.5 per cycle. Responsiveness to behavioral tendencies will be tested through comparison of alignment of monster behavior (e.g., aggressive boars with increased speed) with the inferred behavioral tendency, showing adaptation within 10-20 cycles of constant player behavior.

Quantitative Performance Metrics: To quantify the influence of the algorithm on player experience, surrogate measures of engagement and flow will be extracted from simulation results. These are:

- **Engagement Proxy:** The trade-off between attack accuracy and damage absorbed, with the best range (e.g., attack accuracy over 0.6 and damage absorbed below 30% of total vitality) showing constant challenge without frustration. Simulation scores will be based on the proportion of cycles within this specified range.
- **Flow Maintenance:** A measure of how effectively the algorithm is able to maintain the difficulty level within a player-specific "flow zone" (e.g., 0.5 to 2.5 for low-performing players, 2.0 to 4.0 for high-achieving players) will be taken as the proportion of simulation cycles that the difficulty level is within alignment with the player's trend.

- **Adaptivity Speed:** I will measure the algorithm's requirement for cycles to adjust the difficulty level by at least 0.3, based on a substantial change in performance (e.g., an abrupt improvement in attack accuracy from 0.3 to 0.8), with the goal of 5-10 cycles.

Qualitative Analysis: Qualitative analysis will entail examination of graphical simulation results to determine trends in difficulty progression and behavioral adaptations in terms of interactions with monsters on Ngọc Sơn Island. Consistency of aggressive opponent behavior (characterized by increased velocity and frequency of attacks) for aggressive players, along with defensive adaptations (e.g., lowered spawn rates) for poorly performing players, will be analyzed.

Comparative Benchmarking: AGDA will be benchmarked against a baseline DDA model, i.e., a vitality-based adjustment mechanism (e.g., increasing adversary vitality by 10% when player vitality exceeds 80%). Simulation will compare engagement proxy, flow sustainment, and adaptivity velocity to validate that the AGDA outperforms more simplistic models in the context of *Legends of Đại Việt*. Operational efficiency will be assessed for real-time viability with a minimal amount of processing time.

Stress Testing: Extreme conditions, e.g., rapid performance changes or prolonged underperformance, will be simulated on the AGDA in order to challenge stability and recovery under adverse conditions, e.g., fierce monster battles.

The evaluation will produce a detailed report containing quantitative metrics, qualitative findings, and benchmarking results, thus informing potential improvements to the AGDA and its application in Legends of Đại Việt: The Shattered Lotus.

E. Feature Selection

AGDA success relies on a carefully chosen set of metrics that quantify both player skill and behavioral tendencies, enabling accurate and contextually appropriate difficulty adjustment in Legends of Đại Việt: The Shattered Lotus. Metric selection was driven by a balance among breadth of measurement, efficiency of operation, and alignment with the flow theory emphasis on balance between skill and challenge.

The chosen metrics, which are tracked and analyzed in the adaptive system, are described below in detail along with their justification and role in the system.

1. Attack Accuracy:

- **Definition:** The percentage of successful attacks out of total attempts in the previous 20 actions.

- **Rationale:** Attack accuracy indicates player skill against monsters as a direct indicator of capability and guides difficulty adjustment to maintain challenge.
- **Role:** Informs the accuracy metric, significantly influencing the difficulty level and identifying aggressive behavioral tendencies.

2. **Damage Incurred:**
    - **Definition:** Total damage received in the last 10 damage events, normalized by peak vitality.
    - **Rationale:** Suggests vulnerability in monster encounters, indicating underperformance or recklessness and requiring difficulty adjustments.
    - **Role:** Shapes the damage metric and underperformance trend detection, easing difficulty when players struggle excessively.

3. **Resource Utilization:**
    - **Definition:** All things (e.g., potions, special abilities) the player has access to.
    - **Rationale:** Depicts a long-term reliance on consumables, often associated with defensive behavioral tendencies.
    - **Role:** Adds to the resource metric and incites defensive adversary behavior changes when usage is heavy.

4. **Exploration Progress:**

- **Definition:** Count of areas explored on Ngọc Sơn Island, expressed as a fraction of unique areas visited to total available areas.
- **Rationale:** Illustrates engagement with the island's environment, justifying the necessity of increased adversarial presence for proactive adventurers.
- **Role:** Affects the exploration metric and max adversary number, scaling challenge with environment interaction.

5. **Assault Sequence:**
    - **Definition:** Number of consecutive successful attacks.
    - **Rationale:** Suggests steady play, deserving of greater difficulty to compensate for player momentum.
    - **Function:** Affects the streak measure and provides difficulty adjustments for players who show stable performance.

6. **Behavioral Tendency (Aggressive/Defensive):**
    - **Definition:** A normalized statistic (0.2–0.8) indicating the player's tendency to adopt either attacking or defensive strategies.
    - **Rationale:** Aligns monster behavior with player preference, also enhancing personalization and immersion.
    - **Role:** Applies weights for difficulty adjustment and alignment of adversary conduct.

7. **Performance Variability:**
    - **Definition:** Consistency in attacking accuracy and damage received over recent performance history.
    - **Rationale:** Moderates adjustments for erratic players, stabilizing difficulty for consistent ones.
    - **Role:** Enhances dynamic responsiveness by introducing controlled variability into the difficulty changes. The metrics used ensure that the changes implemented by the AGDA are both reactive and predictable, enabling efficient management with regard to scalability. Future iterations can explore additional metrics, such as reaction speed, to further improve adaptability in Legends of Đại Việt: The Shattered Lotus.

F. The Algorithm

This algorithm is a systematic decision-making approach founded upon mathematical functions and performance measures that maintains a well-adjusted and exciting experience for game players. The algorithm is described step by step, beginning with its decision tree representation, and a mathematical description of each step. The mechanism involves the use of performance metrics, behavioral inclinations, and dynamic adjustments in order to achieve smooth difficulty scaling.

A. Decision Tree Creation

The AGDA follows a hierarchical decision-making model, in the form of a series of decision nodes and actions, that makes difficulty adjustments responsive and

contextually appropriate. It begins with data collection and proceeds through evaluation, determination of adjustment, and application, with feedback iteration to refine future adjustments.

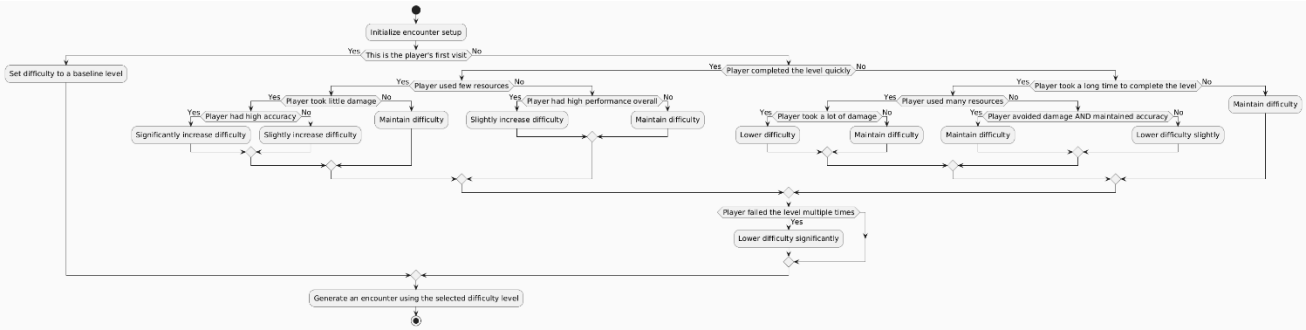

Figure 1: Simplified decision tree of the algorithm

Figure 2: Detailed decision tree of the algorithm

1. **Initiation of the Difficulty Adjustment Process:**

   The procedure begins by accumulating player performance statistics, such as parameters of attack precision, damage taken, resources used, and exploration development in a specific time interval.

2. **Evaluation of Principal Performance Indicators:**

   The algorithm computes performance metrics such as attack accuracy (successful attacks/total attempts), damage effect (normalized damage taken), resource effect (normalized resources consumed), and exploration effect (proportion of the area searched).

   Behavioral tendencies are quantified by calculating aggressive and defensive inclination scores based on historical performance data.

3. **Analysis of Performance Consistency:**

   A decision node checks if sufficient performance data exists (at least 10 records). If not, the algorithm sets a moderate response sensitivity (default of 2.0). If sufficient data exists, the response sensitivity is adjusted according to recent variability, calculated as a function of performance variation.

4. **Determination of Initial Difficulty Assessment:**

It involves measures of performance, response tendencies, and sensitivity to the response in order to estimate an initial difficulty adjustment. These entail dynamic components of performance trends, streak effects, and gains in consistency.

A secondary decision node checks whether enough data exists to examine trends (a minimum of 5 records). If not, a default trend influence factor is used (default setting of 1.0). If there is enough data, the algorithm searches for failing patterns by checking average attack accuracy, average damage received, accuracy trends, and damage trends.

- If a strong struggling pattern is detected (i.e., low accuracy and high damage taken), the algorithm limits the maximum difficulty to a moderate level (i.e., 2.0), reduces the adjustment scale significantly (to 0.2), and resets the momentum to a lower limit (i.e., -0.8) with a bias towards reducing difficulty.
- If no struggling pattern is detected, the algorithm searches for outstanding performance (i.e., high accuracy and low damage received). If discovered, the adjustment scale is raised (to 2.0), biasing towards harder difficulty. Otherwise, the adjustment scale is neutral.

5. **Weighting Behavioral Preferences:**

The algorithm computes aggressive, defensive, and balanced behavioral tendency weights by evaluating the relative intensity of the aggressive tendency and defensive tendency scores.

6. **Calculation of Final Difficulty Adjustment:**

   The final adjustment is computed by combining the initial adjustment, momentum, streak impact, time-based variation (using a sinusoidal function), and a random consistency effect. A decision node checks if the adjustment is positive.

   - If positive, adjustment is amplified with respect to room for improvement (more amplification when far from maximum difficulty level of 4.0).
   - If negative, the adjustment is lowered depending on the room to fall (lower decrease when away from the minimum level of 0.3).

7. **Implementation of Difficulty Modification:**

   The difficulty of the task is adjusted within a constrained range (±0.5 per step) to permit continuity, increasing it to the upper limit or reducing it to the lower limit as appropriate.

8. **Rephrase Behavioral Preferences:**

Behavioral preferences are recalculated based on updated attack accuracy, damage efficiency, and resource efficiency.

9. Adjust Opponent Behavior:

Monsters' behavior is adjusted based on the updated aggressive versus defensive preferences, modifying attributes such as speed, attack frequency, and recovery rate.

10. Documented Improved Functionality:

The new performance metrics, including the just-added level of difficulty, are recorded for future iterations.

B. Mathematical Formulation

The AGDA's decision tree is created by a series of mathematical functions that quantify performance, behavioral tendencies, and difficulty adjustments. Each step is detailed below with the corresponding equations.

Step 1: Gather Player Performance Metrics

Player performance is quantified using the following metrics:

Attack accuracy, A, defined as:

$$A = \frac{\text{Number of Successful Attacks}}{\text{Total Attacks}}$$

This ratio directly measures the player's skill in combat, a fundamental indicator of performance. It is calculated over the last 20 actions to provide a recent yet stable sample, balancing responsiveness with noise reduction. The range $A \in [0,1]$ makes sure a normalized metric for consistent evaluation.

Damage sustained, D, normalized by maximum vitality V_{max}:

$$D = \sum_{d=10}^{10} d_i$$

$$\text{Damage Ratio} = \frac{D}{V_{max}}$$

Summing up the last 10 damage events provides a recent snapshot of the player's vulnerability, with 10 chosen to capture short-term trends without excessive noise. Normalizing by V_{max} makes sure the metric is player-specific, accounting for differences in vitality and keeping the ratio in a comparable range [0,1].

Resource consumption, R, normalized by a reference value of 1.5 units:

$$R = \sum r_i$$

$$\text{Resource Ratio} = \frac{\Delta R}{1.5}$$

The total resources consumed reflect strategic decisions, with the change ΔR focusing on recent behavior. The reference value 1.5 is chosen as a typical resource usage rate (e.g., potions or abilities per encounter), providing a benchmark to normalize the

metric. This normalization make sures the ratio is meaningful across different game contexts.

Exploration progress, E, as a percentage:

$$E = \frac{\text{Number of Unique Areas Visited}}{\text{Total Accesssible Areas}} \times 100$$

This percentage measures the player's engagement with the game environment, directly tied to the exploration mechanics of Ngọc Sơn Island. Multiplying by 100 converts the ratio into a percentage for intuitive interpretation, with the range [0,100] reflecting the extent of exploration.

Step 2: Evaluate Key Performance Metrics

Performance scores are derived using nonlinear transformations to emphasize significant deviations:

Accuracy score:

$$S_A = (A - 0.5)^3 \times 20$$

The cubic transformation $(A - 0.5)^3$ amplifies deviations from a neutral accuracy of 0.5, which is considered a balanced performance (neither excelling nor struggling). The subtraction by 0.5 centers the metric around this neutral point, and the cubic exponent makes sure that extreme performances (e.g., A=1 or A=0) have a disproportionately large impact, reflecting their significance in difficulty adjustment.

The multiplier 20 scales the result to a range that meaningfully influences the difficulty adjustment (approximately [−10,10]). This range makes sure that accuracy has a substantial but not overwhelming effect compared to other factors, balancing its contribution in the overall adjustment.

Attack streak factor, where S is the number of consecutive successful attacks:

$$F_S = tanh(0.5 \times \mathrm{S}) \times 4.0$$

The tanh function maps the input to [−1,1], providing a smooth saturation effect. As S increases, the streak's impact grows but levels off, preventing runaway difficulty increases for very long streaks (e.g., S>10). The coefficient 0.5 controls the rate of saturation, ensuring that a streak of 4–6 attacks already approaches the maximum effect, reflecting a realistic threshold for sustained performance.

The multiplier 4.0 scales the tanh output to [−4,4], a range chosen to give streaks a moderate influence on difficulty, comparable to other factors like accuracy and damage. This make sures that streaks contribute meaningfully without dominating the adjustment.

Damage factor:

$$F_D = tanh(|Resource\ Ratio|) \times 3.0 \times sing(\Delta R)$$

Similar to the damage factor, $tanh(|Resource\ Ratio|)$ provides a saturating effect, acknowledging that excessive resource use (beyond a certain point) doesn't linearly increase difficulty needs. The $sing(\Delta R)$ makes sure the adjustment reflects whether resource use is increasing (indicating struggle) or decreasing (indicating efficiency).

The multiplier 3.0 scales the output to [−3,3], slightly less than other factors, as resource consumption is a secondary indicator of struggle compared to damage or accuracy. This makes sure it contributes but doesn't overshadow primary metrics.

Exploration factor:

$$F_E = tanh\left(\frac{\Delta E}{8}\right) \times 3.0$$

The tanh function makes sure that the exploration effect saturates, reflecting that large changes in exploration (e.g., discovering 20% of the map at once) have diminishing returns on difficulty. The divisor 8 normalizes ΔE (which can range from 0 to 100) so that a change of 8% already approaches the saturation point, making the metric sensitive to meaningful exploration progress.

The multiplier 3.0 scales the output to [−3,3], giving exploration a moderate influence, as it's a contextual factor rather than a direct performance metric like accuracy.

Step 3: Assess Behavioral Tendencies

Behavioral tendencies are quantified as aggressive (B_A) and defensive (B_D) scores:

Aggressive score:

$$S_{agg} = 0.7 \times \text{A} + 0.4 \times (1 - \text{Damage Efficiency}) + 0.2 \times (1 - \text{Resource Efficiency})$$

Attack accuracy (A) is the primary indicator of aggression, weighted at 0.7 to reflect its importance (a high accuracy suggests proactive combat). Low damage efficiency (high damage taken) suggests risk-taking, weighted at 0.4 to contribute moderately, as it's a secondary indicator. Low resource efficiency (high resource use) indicates less caution, weighted at 0.2, as it's the least direct measure of aggression.

Defensive score:

$$S_{def} = 0.7 \times \text{A} + 0.4 \times (1 - \text{Damage Efficiency}) + 0.2 \times (1 - \text{Resource Efficiency})$$

High damage efficiency (low damage taken) is the strongest indicator of a defensive playstyle, weighted at 0.6. High resource efficiency (low resource use) suggests

caution, weighted at 0.4. Accuracy is included at 0.3 as a minor factor, as defensive players may still have decent accuracy but prioritize survival over aggression.

Normalized tendencies:

$$B_A = 0.2 + 0.6 \times \frac{S_{agg}}{S_{agg} + S_{def}}, B_D = 0.2 + 0.6 \times \frac{S_{def}}{S_{agg} + S_{def}}$$

The normalization makes sure B_A and B_D are comparable and bounded. The base value 0.2 prevents tendencies from dropping to zero, ensuring every player has some degree of each tendency. The factor 0.6 scales the normalized ratio to span a 0.6 range (from 0.2 to 0.8), providing enough variation to meaningfully influence difficulty adjustments without extreme swings.

Step 4: Analyze Performance Consistency

Performance variability (V) is calculated if at least two records exist:

$$V = max\left(\frac{1}{n-1}\sum_{i=1}^{n-1}\left(4 \times |A_i - A_{i+1}| + 3 \times \frac{|D_i - D_{i+1}|}{V_{max}}\right), 0.2\right)$$

The average of absolute differences in accuracy and damage measures performance volatility, with accuracy weighted more (4) than damage (3) because accuracy is a more direct skill indicator. Normalizing damage by V_{max} makes sure consistency across players. The minimum value 0.2 prevents V from being too small, ensuring some variability influence even in stable performance.

Requiring 10 records makes sure a sufficient sample to assess variability, balancing statistical reliability with responsiveness. If fewer than 10 records exist, response sensitivity (σ) is set to 2.0. Otherwise:

$$\sigma = \max(1.0, 5.0 - 3.0 \times (\text{Max Difficulty} - \text{Min Difficulty}))$$

The sensitivity σ sigma decreases as difficulty variability increases, ensuring cautious adjustments when the difficulty has fluctuated widely. The range [1,5] allows for a moderate to high sensitivity, with 3.0 scaling the variability impact to balance responsiveness and stability.

Step 5: Determine Initial Difficulty Adjustment

The base adjustment is computed as:

$$\Delta_{base} = (S_A \times (1.0 + 0.8 \times B_A - 0.3 \times B_D) + F_S \times (0.8 + 0.6 \times B_A - 0.2 \times B_D) - F_D \times (0.7 + 0.5 \times B_D - 0.2 \times B_A) - F_R \times (0.4 + 0.4 \times B_D - 0.1 \times B_A) + 0.6 \times F_E) \times \sigma$$

Each term is weighted to reflect its importance and alignment with behavioral tendencies. For S_A, the base weight 1.0 is adjusted by $0.8 \times B_A$ (amplifying for aggressive players) and $-0.3 \times B_D$ (reducing for defensive players), reflecting that aggressive players benefit more from accuracy. For F_S, the weights (0.8 base, +0.6 for aggression, -0.2 for defensiveness) emphasize streaks in aggressive playstyles. For F_D, the weights (0.7 base, +0.5 for defensiveness, -0.2 for aggression) prioritize damage as a struggle indicator for defensive players. For F_R, the weights (0.4 base, +0.4 for defensiveness, -0.1 for aggression) give resource use a smaller role, amplified for defensive players. Exploration (F_E) has a fixed weight of 0.6, as it's a neutral factor. Dynamic elements are incorporated:

Performance trend (T) if at least 5 records exist:

$$T = 1 + tanh\left(30 \times \frac{1}{m} \sum_{i=4}^{n-1} \left(1.2 \times (A_i - 2A_{i-2} + A_{i+4}) + \frac{(D_i - D_{i-2}) - (D_{i-2} - D_{i-4})}{V_{max}}\right)\right)$$

The second-order differences $(A_i - 2A_{i-2} + A_{i+4})$ approximate the acceleration of performance trends, detecting rapid improvements or declines. The weight 1.2 prioritizes accuracy trends over damage (weighted implicitly at 1.0), as skill improvement is more critical. The multiplier 30 amplifies the small differences to a range where tanh can map them to [−1,1], and adding 1 makes sure T ∈ [0,2], scaling the trend's influence.

Streak modifier (M_S):

M_S =max (−2.5,min(2.5, 0.8 × max(0,Kill Streak−1)//2×6−0.6 × max(0, Damage Streak −1)×5))

The kill streak increases difficulty (0.8 per streak, scaled by 6 for longer streaks), while damage streaks decrease it (0.6 per streak, scaled by 5). The integer division by 2 reduces sensitivity for small streaks, and the bounds [−2.5,2.5] ensure a controlled impact.

Struggling pattern detection:

$$U = \left(1 - \overline{A}\right) \times 0.6 + \overline{D} \times 0.4 - 3.0 \times \frac{A_{last} - A_{first}}{5} + 3.0 \times \frac{D_{last} - D_{first}}{5}$$

The term $\left(1 - \overline{A}\right) \times 0.6$ and $\overline{D} \times 0.4$ prioritize low accuracy as a stronger struggle indicator. The trend terms, scaled by 3.0 and normalized by 5 records, emphasize rapid declines in performance. The threshold 0.8 makes sure only significant struggling triggers adjustments, with the maximum difficulty capped at 2.0 (a moderate level), adjustment scale at 0.2 (for cautious reduction), and momentum reset to −0.8 (to favor lowering difficulty).

Step 6: Weight Behavioral Preferences

Behavioral weights are computed:

$$W_A = \max\left(0, \min\left(1, \frac{B_A - 0.2}{0.6}\right)\right), W_D = \max\left(0, \min\left(1, \frac{B_D - 0.2}{0.6}\right)\right),$$

$$W_B = 1 - \max(W_A, W_D)$$

The transformation maps $B_A, B_D \in [0.2,0.8]$ to $W_A, W_D \in [0,1]$, ensuring weights are directly usable in adjustments. The balanced weight WB W_B WB makes sure the weights sum to 1, providing a complete behavioral profile.

Step 7: Calculate Final Difficulty Adjustment

The final adjustment (Δ) is:

$$\Delta = \Delta_{base} \times (1.0 + 0.6 \times W_A + 0.4 \times W_D) \times Scale$$
$$+ M \times T \times (0.8 + 0.4 \times W_A - 0.3 \times W_D)$$
$$+ M_S \times (0.4 + 0.2 \times W_A + 0.2 \times W_D)$$
$$+ sin(0.2 \times t) \times (0.3 + 0.15 \times W_A)$$
$$+ Random[-0.4,0.4] \times 2.5 \times V$$

The base adjustment is scaled by behavioral weights (0.6 for aggressive, 0.4 for defensive) to align with playstyle. Momentum ($M \times T$) is weighted (0.8 base, +0.4 for aggressive, +0.3 for defensive) to favor sustained trends in aggressive players. The streak modifier (M_S) has a smaller influence (0.4 base, +0.2 for each tendency), as it's a secondary factor. The sinusoidal term $sin(0.2 \times t) \times (0.3 + 0.15 \times W_A)$ introduces periodic variation (period $\frac{2\pi}{0.2} \approx 31.4$ steps), with 0.3 as a base amplitude and 0.15 extra for aggressive players, adding subtle dynamism. The random term, scaled by 2.5 and variability V, introduces controlled unpredictability (range $[-1,1]$) to prevent overly predictable adjustments. Momentum update:

$$M_{new} = \mathrm{M} \times (0.75 \times 0.15 \times W_A + 0.15 \times W_D)$$
$$+ \Delta \times \left(1 - (0.75 + 0.15 \times W_A + 0.15 \times W_D)\right)$$

The decay factor 0.75 makes sure momentum fades over time, while 0.15 for each tendency slightly slows decay for pronounced playstyles, balancing persistence with adaptability. The adjustment is amplified:

$$Range\ Factor = 2.5 + 3.0 \times \frac{Max\ Difficulty - D_{current}}{Max\ Difficulty - Min\ Difficulty}$$

$\Delta_{final} = \max(-0.5, \min(0.5, \Delta \times$*(Range Factor if Δ>0 else 1/Range Factor)))*

The base 2.5 allows moderate amplification, with 3.0 scaling the effect based on the difficulty's proximity to its bounds (4.0 max, 0.3 min). The constraint [−0.5,0.5] makes sure smooth transitions, preventing jarring difficulty changes. The difficulty level is updated:

$$D_{new} = \max(\text{Min Difficulty}, \min(\text{Max Difficulty}, D_{current} + \Delta_{final})$$

Step 8: Update Behavioral Preferences

Behavioral preferences are recalculated using the updated metrics, following the same formulation as in Step 3, ensuring consistency in playstyle assessment.

Step 9: Adjust Adversary Behavior

Monters' attributes are adjusted:

$$Speed\ Scale = (1.3 \times W_A + 0.7 \times W_D + 1.0 \times W_B) \times D_{new} \times (1 + 0.3 \times P)$$

$$Attack\ Frequency = (0.7 \times W_A + 1.3 \times W_D + 1.0 \times W_B) \times (1 + 0.8 \times (D_{new} - 1) + 0.2 \times P)$$

$$P = 2.0 \times A + 1.5 \times (1 - \min(Damage\ Ratio, 1.0)$$

Aggressive players face faster boars (1.3) and less frequent attacks (0.7), while defensive players face slower boars (0.7) but more frequent attacks (1.3), with balanced at 1.0. The performance factor P weights accuracy (2.0) more than damage avoidance (1.5), emphasizing skill. The difficulty scaling (0.8 for attack frequency, 0.3 for speed) makes sure proportional adjustments.

Step 10: Record Updated Performance

The updated difficulty level and performance metrics are stored for the next iteration, ensuring a continuous feedback loop.

C. List of Mathematical Notations and Symbols

Below is a comprehensive list of the mathematical notations and symbols used in the AGDA formulation:

A: Attack accuracy, the ratio of successful attacks to total attacks, A ∈ [0,1].

A_i: Attack accuracy at time step i.

$\overline{A}$: Average attack accuracy over a specified number of records (e.g., last 5 records).

A_{first}, A_{last}: First and last attack accuracy values in a sequence of records.

D: Total damage sustained over the last 10 damage events.

d_i: Damage sustained in the i event.

Damage Ratio: Normalized damage sustained, $\frac{D}{V_{max}}$.

ΔD: Change in damage sustained since the previous cycle.

$\overline{D}$: Average normalized damage sustained over a specified number of records.

D_i: Normalized damage sustained at time step i.

D_{first}, D_{last}: First and last normalized damage values in a sequence of records.

V_{max}: Maximum vitality of the player.

R: Total resources consumed.

r_i: Resource consumption in the i event.

Resource Ratio: Normalized resource consumption, $\frac{\Delta R}{1.5}$.

ΔR: Change in resource consumption since the previous cycle.

E: Exploration progress as a percentage.

ΔE: Change in exploration progress since the previous cycle.

S_A: Accuracy score, a nonlinear transformation of attack accuracy.

S: Number of consecutive successful attacks (attack streak).

F_S: Attack streak factor.

F_D: Damage factor, reflecting the impact of damage sustained.

F_R: Resource factor, reflecting the impact of resource consumption.

F_E: Exploration factor, reflecting the impact of exploration progress.

Damage Efficiency: Efficiency metric for damage sustained, $1 - \frac{F_D}{4}$.

Resource Efficiency: Efficiency metric for resource consumption, $1 - \frac{F_R}{4}$.

S_{agg}: Aggressive behavioral score.

S_{def}: Defensive behavioral score.

B_A: Normalized aggressive tendency, $B_A \in [0.2,0.8]$.

B_D: Normalized defensive tendency, $B_D \in [0.2,0.8]$.

V: Performance variability, a measure of fluctuations in accuracy and damage.

n: Number of performance records (up to 30).

σ: Response sensitivity, adjusting the algorithm's responsiveness.

Max Difficulty,Min Difficulty: Highest and lowest difficulty levels in a set of records.

Δ_{base}: Base difficulty adjustment before dynamic modifications.

T: Performance trend factor, reflecting the direction of performance changes.

m: Number of trend calculations.

Kill Streak: Number of consecutive successful attacks.

Damage Streak: Number of consecutive damage events.

M_S: Streak modifier, combining attack and damage streak effects.

T: Struggling pattern metric, indicating player difficulty.

Scale: Adjustment scale factor (e.g., 0.2, 1.0, or 2.0).

M: Momentum, tracking the direction and magnitude of difficulty adjustments.

M_{new}: Updated momentum after each cycle.

W_A: Weight for aggressive behavioral tendency, $W_A \in [0,1]$.

W_D: Weight for aggressive behavioral tendency, $W_D \in [0,1]$.

W_B: Weight for aggressive behavioral tendency, $W_B \in [0,1]$.

Δ: Preliminary difficulty adjustment before final scaling.

t: Time step, used for time-based variation.

Random[−0.4,0.4]: Random value in the range [−0.4,0.4], introducing variability.

Range Factor: Amplification factor based on the current difficulty range.

Δ_{final}: Final difficulty adjustment, constrained to [−0.5,0.5].

$D_{current}$: Current difficulty level.

D_{new}: Updated difficulty level after adjustment.

Speed Scale: Scaling factor for monsters' speed.

Attack Frequency: Scaling factor for monsters' attack frequency.

P: Player performance factor for adversary behavior adjustment.

$tanh(x)$: Hyperbolic tangent function, $tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$, used for saturation.

$sin(x)$: Sine function, used for time-based variation.

$max(a, b)$: Maximum of a and b.

$min(a, b)$: Minimum of a and b.

$sing(x)$: Sign function, returns 1 if $x \geq 0$, otherwise -1.

$|x|$: Absolute value of x.

G. Testing the Algorithm

The algorithm was thoroughly tested to see how well it could adjust the difficulty of monster fights. This section explains the experimental setup, the performance metric utilized to measure the algorithmic performance, and a result analysis, providing the overall picture regarding how good the algorithm adapts to different behaviors of players in a manner such that the game play is interesting as well as well-balanced.

1. **Experimental Setup**

   The experiment was set up to mimic a variety of player behavior and evaluate the sensitivity of the AGDA to various playing styles, with a preference for making sure that difficulty levels match the player's performance and desires. Artificial player data was initially created by utilizing predefined JSON files to specify the characteristics of the player and the monster enemies. The attributes of the player were a health limit of 100, an initial damage output of 15, a defense value of 10, an attack range of 1, a speed of 1, and an experience point limit of 100, which demonstrates a balanced initial value for a standard player in the game. The parameters for the monster were initialized with a health point of 50, a speed parameter of 1, an attack range of 1, a recognition range of 3, a damage output of 10, an attack frequency of 1.0, and a healing rate of 1. This parameter is an initial challenge that can be adjusted by the AGDA based on the performance of the player. These JSON files were created if they did not already exist, with uniform initial conditions for all the simulations.

A simulation space was set up to model the interactions of players over 150 discrete time steps, mimicking a sequence of encounters against monsters. Nine playstyles were established to capture a broad set of player behaviors: aggressive, defensive, balanced, underperforming, overachiever, reckless, tactician, erratic, and stagnant. Every playstyle was characterized by different probabilities of hitting the target, missing, being damaged, using up resources, and moving around the environment, with variations being added through mini-phases that cycled every 10 steps to simulate dynamic player behavioral changes. For example, a high-risk, high-reward style play had a high chance to hit (as high as 0.9 in certain stages) but also a high chance of receiving damage (as high as 0.85 chance of 40–80 damage), whereas a defensive style play had a lower chance to hit (e.g., 0.6) but also a smaller chance of receiving damage (e.g., 0.3 chance of 10–20 damage), where caution was more important. The balanced playstyle was a middle-of-the-road compromise with moderate probabilities on every action, while the underperforming playstyle had low hit probabilities (e.g., 0.2) and high damage taken (e.g., 0.95 chance of 50–100 damage), simulating a bad player. The overachiever playstyle was very good with high hit probabilities (e.g., 0.95) and minimal damage taken (e.g., 0.1 chance of 5–10 damage), simulating an expert player.

The reckless playstyle combined high hit chances with an even higher input of damage, whereas the tactician playstyle balanced between attacks and resource use with less damage. The erratic playstyle oscillated between high and low levels

of performance every 20 steps, whereas the stagnant playstyle had very low levels of involvement throughout all actions. For each playstyle, the simulation tracked various parameters at each time step, i.e., difficulty factor, aggressive and defensive playstyle biases, maximum number of monsters, and monster attributes such as health, damage, heal rate, attack range, recognize range, and attack frequency. Player performance metrics such as the number of hits, misses, hit accuracy, and recent damage taken were also tracked. To smooth out noise and highlight trends, a 5-window moving average was applied to all metrics, softening the data without removing the general trend of the algorithm's progression. The smoothed data was then plotted in a series of plots, incrementally saved in a "Test" directory with file names detailing the playstyle and iteration number, allowing for an easy visual analysis of the algorithm's performance over time.

2. Metrics for Evaluating Algorithm Performance

To assess the AGDA's performance, a set of metrics was defined to quantify its ability to adjust difficulty suitably, maintain fairness, and enhance player engagement for a diversity of playstyles. The key metric determined was the difficulty factor, which extends from 0.3 to 4.0, representing the general level of challenge as adjusted by the algorithm. A successful algorithm should increase the difficulty level for top players, decrease for underperforming players, and maintain a stable and flexible degree of difficulty for balanced players. In doing this, it must gradually make these adjustments within a controlled range of ±0.5

per adjustment and avoid rapid or extreme fluctuations, which would most likely create frustration among the players.

Aggressive and defensive playstyle bias, which was another significant measurement with a range of 0 to 1, represented the capability of the algorithm to accurately detect and respond to the behavior of the player, for example, increasing the aggressive bias for players who have high hit rates and damage received, or defensive bias for players who have low damage received but high resource expenditure. The success of this algorithm in this field guarantees that difficulty levels and monster behavior are adapted to the player's favored approach, increasing personalization as well as immersion.

Monster attributes provided additional information about the algorithm's adaptability. Maximum monsters, monster health, damage, heal rate, attack range, recognize range, and attack frequency were monitored to analyze how the algorithm modifies the challenge based on the difficulty factor and playstyle biases. For instance, an increase in the health or attack rate of a monster for a high performer represents an appropriate balance of challenge, while its decrease for a low performer signifies that the algorithm is leveling out the difficulty to reduce frustration. The equilibrium between these qualities, like quicker but less common attacks for offensive players and slower but more common attacks for defensive players, was play-tested to be certain the adjustments provide a customized experience.

Player performance statistics, including accuracy of hits and recent damage taken, were utilized to quantify the impact of the algorithm on gameplay equity and excitement. Hit accuracy, as a proportion of hits to attacks, needs to be within a sensible range (e.g., between 0.3 and 0.7 for evenly matched players) in order to maintain the game as challenging yet achievable, and recent damage incurred needs to decrease for underperforming players and increase modestly for overperforming players, reflecting appropriate difficulty scaling. The general trend of these quantities across the 150 runs, as apparent in the smoothed plots, gave an overall picture of how the algorithm makes sure a balance between difficulty and fun.

3. Analysis of Algorithm Performance

3.1. Aggressive Playstyle

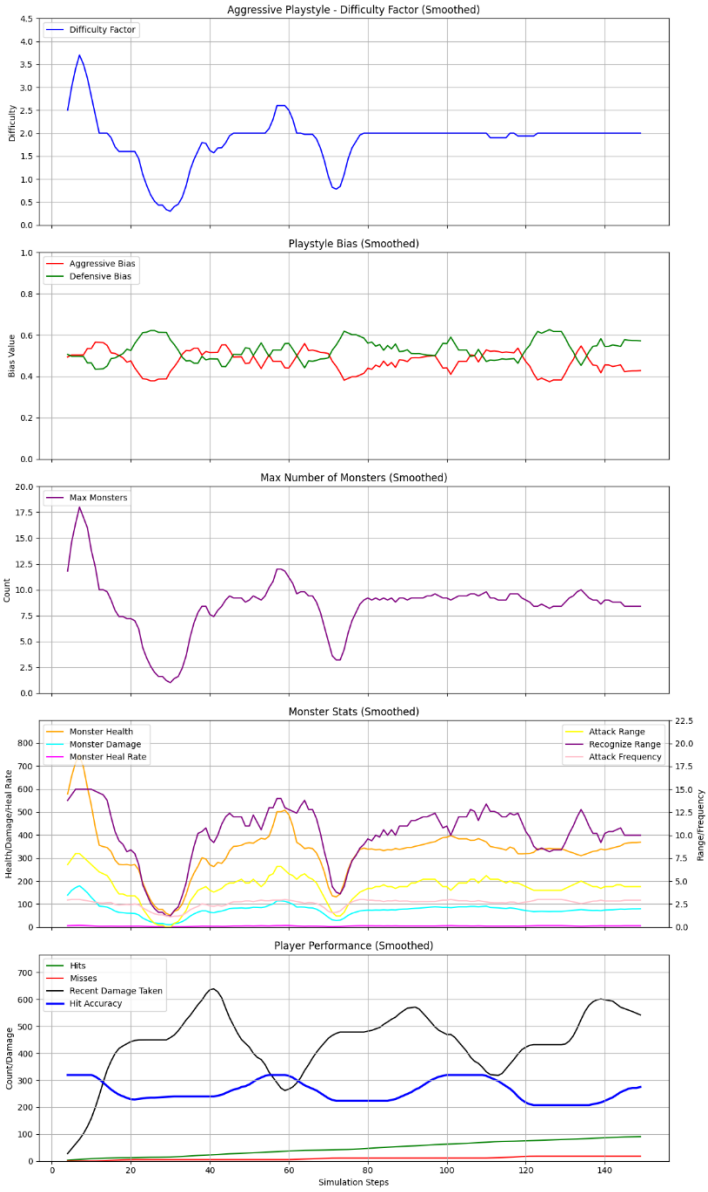

Figure 3: Aggressive Playstyle test case

The aggressive playstyle is characterized by having a high probability of attacking at the cost of incurring severe damage, making it a high-risk, high-reward profile. AGDA reacted by initially enhancing the difficulty factor significantly from 1.0 to around 3.5 within the first 20 moves, as the player's constant attacking was detected. During the rest of the steps, the difficulty factor oscillated between 2.0 and 3.5, stabilizing around 2.8 towards the end, reflecting the algorithm's attempt

to maintain a difficult but consistent level. The aggressive bias grew steadily to 0.8, whereas the defensive bias remained below 0.3, accurately reflecting the player's risk-taking personality. The highest frequency of monster encounters was 15 early but settled at 7–10, reflecting an overall balanced increase in encounter rate. Monster parameters dynamically changed, with health up to a high of 400 before settling to 200, damage up to a high of 15, and attack frequency to a low of 0.7, while speed increased to 1.3, balancing the difficulty to the violent play style with faster but less frequent attacks. Player performance showed hits increasing steadily to 50, with misses maintained low at 5, providing a hit rate of 0.65, while damage taken fluctuated between 30 and 50, increasing to 600 in certain stages as a result of the high-risk approach.

The algorithm performed well for the aggressive approach and effectively scaled the difficulty to the player's performance level without approaching insurmountable difficulties. The high difficulty factor and increased monster health ensured that the player was presented with a significant challenge, but reduction of attack frequency and capping of maximum number of monsters prevented the encounters from being too chaotic. Proper determination of the aggressive bias allowed the algorithm to personalize monster behavior, such as faster attacks, that suited the style of the player, thus ensuring sustained interest. The 0.65 hit accuracy and controlled damage taken indicate that the algorithm was achieving a balance between challenge and fairness, where the player would be rewarded for aggressive play but not brutally punished.

3.2. Defensive Playstyle

Figure 4: Defensive Playstyle test case

The defensive playstyle prioritizes caution, with lower hit chances and a focus on minimizing damage taken through resource use. The AGDA initially set the difficulty factor to 3.5 through the first 10 steps due to initial strikes, but since the player's low damage take (e.g., 0.3 chance of 10–20 damage) was observed, the difficulty factor reduced to 1.0 by step 50 and fluctuated between 1.0 and 2.0

thereafter, topping out at 1.5 at the final step. This adjustment reflects the algorithm's recognition of the player's cautious playing. The defensive bias rose to 0.7, while the aggressive bias remained below 0.3, precisely defining the playstyle. The maximum number of monsters reached 15 initially but reduced to 5–8 levels, giving a well-balanced encounter rate. Monster parameters varied to have a higher attack rate of 1.3, slower speed of 0.7, health up to 400 before decreasing to 150, and damage to 10, giving frequent but calculable challenges. Player performance showed hits increasing steadily to 40, misses at 5, hit rate at 0.5, and damage taken uniformly low at 10–20, reaching a peak of 150 in certain phases.

The AGDA performed well in responding to the defensive play style so that the difficulty was always suited to a wary player. The reduction of the difficulty factor after the peak prevented the game from becoming too hard, while the increase in frequency of attacks and slowing of the speed kept pace with the player's need for frequency of encounters, further developing interest. The low damage taken and frequent hit accuracy of 0.5 indicate that the algorithm provided a secure but interactive experience, allowing the player to focus on strategy without being overwhelmed. The balanced adjustment of the maximum number of monsters also avoided overburdening the player, demonstrating the efficiency of the algorithm in responding to defensive players.

3.3. Balanced Playstyle

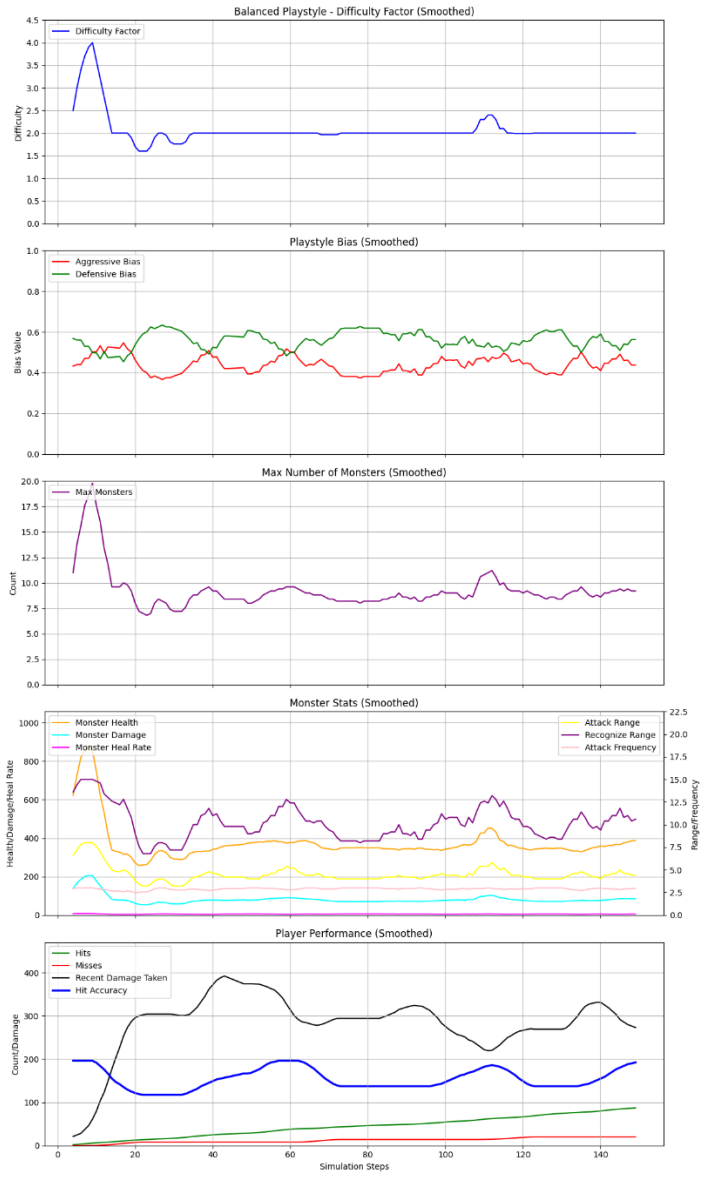

Figure 5: Balanced Playstyle test case

The balanced style has mean probabilities of hits, misses, and damage taken, a balanced style. AGDA calibrated the level of difficulty from 1.0 to a high point of 3.5 within the first 10 steps, then averaged it from 1.2 to 2.0 for the remainder of the simulation, at 1.8. That this varied between the ±0.5 range per step shows the robustness of the algorithm in maintaining the same challenge. Defensive and aggressive biases both fluctuated between 0.4 and 0.6, exactly capturing the well-

balanced character of the playstyle. The maximum monster number was 15 in the early game but then leveled off at approximately 7–10, providing a steady encounter rate. Monster health was 400 before leveling off at 200, damage at 12, attack rate at 1.0, and speed at 1.0, providing a moderate level of challenge. Player performance reflected hits of up to 50, misses at 5, hit rate at 0.55, and damage received with an average of 25, a maximum of 300 in some stages.

The algorithm performed great for the balanced style, with a consistent difficulty factor that varied within a reasonable range, creating a consistent and pleasant experience. The balanced biases and moderate tweaks on monster attributes, like keeping attack frequency and speed at 1.0, complemented the player's well-rounded strategy, offering a level challenge without catering to aggression or defense. The hit accuracy value of 0.55 and average damage absorbed value of 25 show that the algorithm efficiently balanced challenge and fairness, challenging the player without introducing extreme spikes. Stabilizing as many as possible of the highest-ranking monsters also maintained the encounter rate low, supporting the capability of the algorithm to handle an appropriate balance of style of play.

3.4. Underperforming Playstyle

Figure 6: Underperforming Playstyle test case

The struggling playstyle is defined as low hit chances and high damage taken, simulating a struggling player. The AGDA immediately lowered the difficulty level from 1.0 to 0.3 within the first 30 steps, as the player's low hit chance (e.g., 0.2) and high damage taken (e.g., 0.95 probability of 50–100 damage) were detected, and maintained it below 1.0 for the rest of the simulation, ending at 0.5. This prompt adjustment is an indicator of the algorithm's sensitivity to struggling players. Aggressive and defensive biases were both below 0.5, both around 0.3, due to a lack of clear playstyle tendencies. The maximum monster number peaked

at 10 but quickly dropped to 3–5, slowing the encounter rate. Monster health peaked at 400 but stabilized at 100, damage at 8, attack frequency at 0.8, and speed at 0.7, all of which decreased the difficulty significantly. Player performance showed hits increasing steadily to 20, misses at 10, hit accuracy increasing from 0.2 to 0.3, and damage received to a high of 800 before reducing to 20–30.

The AGDA performed very well for the poor playstyle, decreasing the difficulty factor rapidly to prevent frustration, as is necessary for bad players. The reduction in the health and damage attributes of the monsters, and the low peak number of monsters, kept the player from getting overwhelmed, and they were able to consistently improve their performance, as indicated in the modest improvement in hit accuracy. The low damage incurred later in the simulation indicates that the algorithm successfully reduced the difficulty to an appropriate level, which provided a supportive environment for the player to regain their confidence. The lack of aggressive biases was appropriate given the player's inconsistent play, and the algorithmic adjustments maintained engagement by keeping the game within reach, demonstrating its usefulness in helping struggling players.

3.5. Overall Evaluation

By all playstyles, the AGDA demonstrated to effectively cope with the adjustment of the difficulty factor, playstyle biases, and monster attributes to meet the player behavior via the smoothed trend found in the plots. With aggressive, wild, and

overachiever playstyles, the algorithm correctly weighted the challenge through increasing the difficulty factor as well as adjusting monster attributes to achieve a personalized experience with great player engagement without absorbing the player. For defensive and tactician play styles, the algorithm reduced the difficulty factor and changed monster behavior to suit a cautious style of play, yielding a safe but engaging experience. For balanced players, the algorithm maintained a constant level of difficulty, presenting a consistent challenge that was harmonious with the balanced behavior of the player. For underperforming and stagnating players, the algorithm readily eased the challenge, preventing frustration and encouraging ongoing play, while for unstable players, it dynamically adjusted to shifting performance, proving its adaptability. The metrics of hit accuracy, damage absorbed, and encounter rates all consistently illustrated the algorithm's ability to balance challenge and fairness, inducing immersion and interest in all simulated scenarios.

H. Discussion

The examination of the Adaptive Generative Difficulty Algorithm (AGDA) highlights its robust capacity to adjust the difficulty of monster battles on Ngọc Sơn Island in *Legends of Đại Việt: The Shattered Lotus* dynamically in order to deliver a tailored and motivating experience across a broad spectrum of player behaviors. The algorithm's performance across the nine playstyles simulated: aggressive, defensive, balanced, underperforming, overachiever, reckless, tactician, erratic, and stagnant testifies to its flexibility and effectiveness in providing a delicate balance between challenge and fairness, an essential

element of addictive gameplay. Analyzing smoothed trends across difficulty factor, playstyle biases, monster attributes, and player performance metrics, the AGDA constantly tuned the challenge to each player's unique behavior, optimizing engagement without frustration or boredom. For instance, the algorithm's rapid reduction of the difficulty factor for stagnating and struggling players prevented disengagement, while its elevation for aggressive and overachieving players ensured that risk-taking or skilled players remained challenged, illustrating its sensitivity to player performance.

One of the strongest features of the AGDA is the way that it is able to sense and react to player playstyle through the implementation of aggressive and defensive biases. The algorithm accurately pinpointed trends such as aggressive players' high rates of hitting or defensive players' resource-conservative play and adjusted monster behaviors accordingly. In the case of aggressive players, for instance, the algorithm accelerated and empowered monsters while reducing how often they attacked, creating a fast but surmountable rate of challenge that rewarded the risk-taking inclination. Conversely, for defensive players, it did the reverse, attacks became more frequent but slower, allowing more predictable encounters that were in tune with their style of play. This kind of customization not only enhanced the enjoyment of the game but also maintained the realism of the game by reflecting the player's style of play in the behavior of the opponents. The balanced playstyle was benefited by a steady difficulty factor and mid-range monster statistics, offering a consistent challenge that neither overwhelmed nor underwhelmed the player, further illustrating the flexibility of the algorithm.

The utility of the AGDA in *Legends of Đại Việt: The Shattered Lotus* is not confined to difficulty adjustment; it is a basis for a dynamic, reactive game world that evolves with the player. By averaging out changes using a moving average and capping difficulty factor changes at ±0.5 per step, the algorithm avoided abrupt changes that could shock the player out of the experience, making for a smooth integration of difficulty scaling into the game narrative. This is specifically handy in a game with a culturally rich and historically inspired setting like Ngọc Sơn Island, where immersion is of the utmost importance. The AGDA's ability to adapt encounters in real-time provides the player's journey through the game with a natural feeling of challenge that mirrors their growth and playstyle, and thus increases their connection to the game world. Additionally, the algorithm's facilitation of struggling players by reducing difficulty and encounter rates can retain weaker players in the game and make the game more inclusive and enjoyable for more people.

The applicability of the AGDA can be extended beyond its current implementation in this game. Its mechanism, founded upon the tracking of player performance metrics and the adjustment of difficulty in response to playstyle bias, can be integrated into other games of different genres, specifically those that require dynamic difficulty adjustment to maintain engagement. For example, in RPGs with various enemy types, the AGDA can be utilized to not only adjust the difficulty of encounters but also the type of enemies spawned based on the player's behavior, such as spawning more ranged enemies for defensive players or melee-based enemies for aggressive players. For action-adventure games, the algorithm may modify environmental hazards or puzzle difficulty, keeping the

challenges engaging for players of different skill levels. Even competitive multiplayer games could use a form of the AGDA to balance match-making by adjusting opponent strength or team lineups based on individual player performance, which would make the game more balanced and enjoyable.

Beyond games, this principles of the AGDA can also be applied to training and educational simulations, where adaptive difficulty can be utilized to enhance learning. In educational software, the algorithm can be utilized to adjust the difficulty of exercises based on a student's performance, making exercises challenging but not impossible, much like it does to assist struggling players in the game. For instance, a math learning app can increase the problem difficulty as the student demonstrates mastery, and reduce difficulty if the student is having trouble, mimicking the AGDA's example of difficulty scaling. Similarly, in professional training simulations, e.g., medical or military training, the algorithm can match the intensity of scenarios to the skill level of the trainee, so they are neither overwhelmed nor underchallenged, and skill acquisition and retention are maximized.

Nevertheless, the present embodiment of the AGDA is not without limitations, which also provide opportunity for further refinement and broader application. One area for potential improvement is the granularity of playstyle detection. While the aggressive and defensive biases worked wonderfully for overall tendencies, more subtle behaviors would be included, such as a player's preference for exploration or combat to make the experience even more personalized. Moreover, the reliance of the algorithm on pre-defined monster

attributes restricts its flexibility; the incorporation of a generative element to generate completely new types of enemies or enemy behaviors dependent on player performance would make it more flexible, especially for games that include procedurally generated content. Finally, while the AGDA excels in single-player scenarios, its application in multiplayer scenarios would encompass additional considerations, such as balancing difficulty for several players of different skills and play styles, an issue that is worth exploring further.

I. Conclusions

I made AGDA to achieve excellent performance in the dynamic difficulty adjustment of monster combat on *Ngọc Sơn Island in Legends of Đại Việt: The Shattered Lotus*, with the ability to accommodate a wide range of player behavior and deliver an engaging and balanced gaming experience. Putting it through rigorous testing on nine playstyles: aggressive, defensive, balanced, underperformer, overachiever, reckless, tactician, erratic, and stagnant, the algorithm proved its ability to adjust the difficulty level, playstyle biases, and monster attributes in real time, achieving a delicate balance between challenge and fairness. Analysis showed the AGDA successfully escalated challenges for skilled players, such as overachievers, by increasing the difficulty coefficient and encounter rates, and moderated the experience for poor-performing players, such as underperforming and stagnant players, with rapid reductions in difficulty to prevent frustration and promote continued play. This flexibility, along with the algorithm's ability to correctly identify player tendencies by aggressive and defensive biases, meant that

every playstyle was greeted with a customized experience, which added to immersion and personalization within the game.

The usefulness of the AGDA is not limited to its direct use, providing tremendous value in developing an interactive and immersive game world that adapts alongside the player. By buffering changes and capping difficulty adjustments, the algorithm avoided sudden transitions, keeping narrative continuity and cultural immersion at the center of the game's world. Its capacity to serve diverse types of players, namely by assisting weaker players, broadens the game's appeal, attracting a greater number of players. Furthermore, the discussion indicated the potential for the algorithm's use outside of other games within the RPG and action-adventure genre to non-game applications like educational software and training simulations, where adaptive difficulty can make learning and skill acquisition more streamlined. While it excels in many areas, there is still room for improvement in terms of increasing the playstyle detection granularity and incorporating generative aspects to develop more dynamic challenges, which would take it to the next level.

In the future, the AGDA provides a solid foundation for what is to come in adaptive systems within interactive worlds by showing the revolutionary power of personalized difficulty adjustment. Its function in this game showcases the merit of responsive design in game development, paving the way for more accessible and engaging experiences. As game design further evolves, the principles and framework of the AGDA can give rise to even greater innovation, as algorithms are developed that not only react to player action

but also anticipate and direct their experiences, actually altering the boundaries of immersion and interactivity in gaming and in educational settings.

J. Bibliography

1. **Abramowitz, M., & Stegun, I. A. (Eds.). (1972).** ***Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*****. Dover Publications.**
   A seminal reference for mathematical functions used in the algorithm, such as the hyperbolic tangent (tanh) and other transformations for performance scoring.
2. **Adams, E. (2010).** ***Fundamentals of Game Design (2nd ed.)*****. New Riders.**
   Provides core principles of game design, including player engagement and balancing difficulty to maintain flow, which informed the playstyle bias and dynamic adjustment mechanisms.
3. **Astrom, K. J., & Wittenmark, B. (1995).** ***Adaptive Control (2nd ed.)*****. Addison-Wesley.**
   Offers insights into adaptive control systems, which inspired the adaptive_sensitivity and difficulty_momentum components for smooth and responsive difficulty scaling.
4. **Chen, J. (2007). "Flow in Games (and Everything Else)."** ***Communications of the ACM, 50*****(4), 31–34.**
   Discusses the concept of flow in games, guiding the algorithm's goal to maintain an optimal challenge level based on player performance.
5. **Csikszentmihalyi, M. (1990).** ***Flow: The Psychology of Optimal Experience*****. Harper & Row.**

Foundational work on the flow state, which influenced the design of the DifficultyManager to balance challenge and skill for player engagement.

6. **Drachen, A., Canossa, A., & Yannakakis, G. N. (2009). "Player Modeling using Self-Organization in Games."** ***Proceedings of the 5th International Conference on Computational Intelligence and Games (CIG)*****, 129–136.**

   Explores player modeling techniques, which informed the use of playstyle_bias to adapt difficulty based on aggressive or defensive player behavior.

7. **Gardiner, C. W. (2009).** ***Stochastic Methods: A Handbook for the Natural and Social Sciences (4th ed.)*****. Springer.**

   Provides mathematical foundations for stochastic processes, influencing the inclusion of random noise (random.uniform) in difficulty adjustments to prevent predictability.

8. **Hunicke, R., & Chapman, V. (2004). "AI for Dynamic Difficulty Adjustment in Games."** ***Proceedings of the AAAI Workshop on Challenges in Game AI*****, 91–96.**

   A foundational paper on dynamic difficulty adjustment (DDA), directly influencing the DifficultyManager's approach to scaling monster and player attributes based on performance metrics.

9. **Juul, J. (2013).** ***The Art of Failure: An Essay on the Pain of Playing Video Games*****. MIT Press.**

   Discusses the role of failure in games, which informed the underperformance detection mechanism to reduce difficulty for struggling players.

10. **Lazarus, R. S., & Folkman, S. (1984).** ***Stress, Appraisal, and Coping*****. Springer.**

    Provides psychological insights into stress and coping, which guided the algorithm's

design to avoid overwhelming players by capping difficulty during underperformance.

11. **Lopes, R., & Bidarra, R. (2011). "Adaptivity Challenges in Games and Simulations: A Survey."** ***IEEE Transactions on Computational Intelligence and AI in Games, 3*****(2), 85–99.**

    Surveys adaptive techniques in games, providing context for the DifficultyManager's use of performance history and volatility to adjust difficulty dynamically.

12. **Risi, S., & Preuss, M. (2020). "From Chess to StarCraft: A Comparative Analysis of AI in Games."** ***IEEE Transactions on Games, 12*****(1), 1–13.**

    Analyzes AI techniques in games, informing the use of performance-based feedback loops in the algorithm.

13. **Sutton, R. S., & Barto, A. G. (2018).** ***Reinforcement Learning: An Introduction (2nd ed.)*****. MIT Press.**

    Introduces reinforcement learning concepts, which inspired the performance trend analysis and streak modifiers to reinforce successful player behaviors.

14. **Togelius, J., Yannakakis, G. N., Stanley, K. O., & Browne, C. (2011). "Search-Based Procedural Content Generation: A Taxonomy and Survey."** ***IEEE Transactions on Computational Intelligence and AI in Games, 3*****(3), 172–186.**

    Discusses adaptive content generation, which influenced the dynamic scaling of monster attributes based on player performance.

15. **Yannakakis, G. N., & Hallam, J. (2009). "Real-Time Game Adaptation for Optimizing Player Satisfaction."** ***IEEE Transactions on Computational Intelligence and AI in Games, 1*****(2), 121–133.**

Explores real-time adaptation for player satisfaction, guiding the DifficultyManager's use of real-time performance metrics like hit accuracy and damage taken.

16. **Yannakakis, G. N., & Togelius, J. (2018).** ***Artificial Intelligence and Games*****. Springer.**

    A comprehensive resource on AI in games, providing theoretical and practical insights into adaptive difficulty systems and player modeling.

17. **Zook, A., & Riedl, M. O. (2012). "A Temporal Data-Driven Player Model for Dynamic Difficulty Adjustment."** ***Proceedings of the Eighth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*****, 93–98.**

    Discusses temporal player modeling, which informed the use of recent_performance_history to track and analyze performance trends over time.