

## Article

# A Modular Framework for Dynamic Difficulty Adjustment in Boss Battles: A Case Study in The Elder Scrolls V: Skyrim

Raul Brumar <sup>1,\*</sup>, Kevin Jacob Stanly <sup>2</sup> and Stelian Nicola <sup>1</sup><sup>1</sup> Department of Automation and Applied Informatics, Politehnica University Timișoara, 300006 Timișoara, Romania; stelian.nicola@upt.ro<sup>2</sup> Department of Computer and Information Technology, Politehnica University Timișoara, 300006 Timișoara, Romania; kevin-jacobs.stanly@student.upt.ro

\* Correspondence: raul.brumar@upt.ro

## Abstract

Dynamic Difficulty Adjustment (DDA) is a technique in game design used to adapt challenge in real-time based on player performance with the goal of maintaining engagement. Existing DDA implementations are predominantly tied to specific game engines and rely on invisible static scaling. This paper presents a modular engine-independent DDA framework for boss battles that produces observable behavior adaptations. The framework integrates live telemetry of combat with an external Python-based adaptive controller. A dominance ratio difficulty model with exponential smoothing drives one-directional phase transitions that trigger behavioral changes without modifying the core game engine. The framework was evaluated through an exploratory study with 30 participants from varying skill levels completing 166 boss fight sessions in Skyrim. Post-test analysis combined self-reported Likert-scale survey data with quantitative K-means clustering on seven telemetry features yielding four unique profiles: Reckless, Struggling, Defensive and Dominant. Survey results indicated strong self-reported player engagement (96.7%), high perceived fairness (72.4%) and near-unanimous behavioral awareness (93.3%). An individual trace analysis illustrated that the DDA made contextually appropriate decisions across the observed sessions. These results demonstrate that modular, behavioral and transparent DDA are practically achievable and positively received with implications for adaptive game design beyond any single engine or title.

**Keywords:** dynamic difficulty adjustment; adaptive gameplay; player modeling; K-means clustering; game AI; Papyrus scripting; boss battle design; flow theory; behavioral adaptation; player experience

Academic Editors: Roberto Ribeiro and Anabela Marto

Received: 23 June 2026

Revised: 22 July 2026

Accepted: 27 July 2026

Published: 29 July 2026

**Copyright:** © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

## 1. Introduction

Game difficulty is one of the most significant design decisions in games or any interactive entertainment. A game that feels too easy fails to engage skilled players and a game that is too hard distances newcomers and casual players. Static difficulty modes, in most games typically labeled Easy, Normal and Hard, represent an uneven and fixed solution to a fundamentally dynamic problem [1].

Dynamic Difficulty Adjustment (DDA) addresses this limitation by monitoring player performance in real time and adjusting game parameters or behaviors accordingly. The concept is grounded in Csikszentmihalyi's flow theory [2,3] that describes a psycho-

logically optimal state in which perceived challenge and skill demonstrated remain in equilibrium. When challenge exceeds skill, players experience frustration and when skill surpasses challenge, it results in boredom. DDA aims to maintain players within this “flow channel” throughout the sessions [4].

Boss battles in role-playing games present a particularly compelling domain for DDA. Unlike open-world exploration or random encounters, boss fights provide well-controlled and high-stakes sessions that are best suited for telemetry collection and adaptive response. Existing DDA implementations in commercial games and adaptive systems share two notable limitations—they are coupled to their host game engine and they rely on invisible statistical scaling (e.g., adjusting health or damage multipliers) that players cannot observe or connect to their own performance [5–8].

This paper presents a modular DDA framework for boss battles designed to tackle both limitations. The framework decouples adaptive logic from the game engine entirely, communicating through a shared file interface and produces observable phase-based adaptation in the boss rather than hidden parameter scaling. The Elder Scrolls V: Skyrim (hereafter “Skyrim”) serves as a case study for implementation and empirical evaluation.

The main contributions of the work include:

1. A modular engine-independent DDA architecture communicating through a shared file interface.
2. A real-time dominance ratio difficulty model with exponential smoothing producing stable difficulty trajectories.
3. A three-phase behavioral adaptation system implemented in Papyrus scripting.
4. An empirical evaluation with  $n = 30$  participants combining qualitative survey data and K-means player profiling.

## 2. State of the Art

DDA has been studied from different angles including foundational theoretical frameworks, commercial game implementations, machine learning controllers, player profiling methods and questions of adaptive transparency. The following subsections cover each category and gap addressed by the present work.

### 2.1. Theoretical Foundations

Csikszentmihalyi’s flow theory [2,3] states that the optimal engagement comes into play when the level of perceived challenge is equal to that of an individual’s perceived skills. Studies in gaming contexts have largely confirmed this association [4,9]. Sweetser and Wyeth [10] translated flow in games into the GameFlow model and a set of measurable criteria: concentration, challenge, player skills, control, clear goals, feedback, immersion and social interaction that have been utilized extensively to assess DDA systems. Sherry et al. [11] also found that challenge was the strongest predictor of time spent playing video games regardless of gender or game genre, highlighting the importance of difficulty tuning for player retention. More recently, research has focused on challenge/enjoyment in FPS contexts [12], validating that self-rewarding, balanced challenge is the primary factor for a positive gaming experience.

### 2.2. Classic and Commercial DDA Systems

The concept of run-time difficulty scaling can be traced back to early arcade hardware. Hunicke [13] reinforced DDA as a design tool and articulated the theoretical case for adaptive difficulty in games. Spronck et al. [14,15] introduced dynamic scripting, which is a reinforcement learning technique where the rule weights are adjusted online based on performance-based metrics, demonstrating that adaptive AI can improve enter-

tainment value without reducing perceived fairness. Their work remains the foundational reference for performance-based adaptation.

Among commercial game implementations, *Resident Evil 4* (Capcom, 2005) [16] implements an Adaptive Game Script (AGS) that constantly monitors a composite score derived from player ammunition, health, inventory states, and recent deaths, and adjusts enemy health, movement speed, spawn rate and resource drop rates. All adjustments are invisible to the player and strictly embedded in the game's C++ engine, making the system non-transferable [17]. *Left 4 Dead* (Valve, 2008) [18] introduced an AI Director that monitors aggregated team performance and adjusts spawn rates and timing. The Director prioritized pacing over runs rather than reacting to instantaneous metrics, a principle reflected in the cooldown and minimum sample mechanism of the present framework. Like the AGS, it is engine-coupled and only targets cooperative play rather than single-player boss battles.

### 2.3. Academic DDA Research

Andrade et al. [19] is among the few academic DDA works applied to RPG combat. Their rule-based controller re-balances enemy statistics in a turn-based prototype between combats. A controlled evaluation demonstrated statistically significant increases in fairness perception and enjoyment over static difficulty, and is the closest academic precedent to this work. Andrade et al. employed a static control condition—the “gold standard”, which is absent in this work and identified as the main direction of future work.

Yannakakis and Togelius [20] proposed experience-driven procedural content generation (EDPCG), a conceptual model in which content is generated according to maximizing modeled player affect. Their research establishes the theoretical relationship between player modeling and adaptive content and has gone on to many more games domains. Pedersen, Togelius and Yannakakis [21] utilized player experience modeling in *Super Mario Bros.*, showcasing that even brief gameplay sessions provide enough information to predict player preferences, a notion that the current framework's two-second telemetry sampling interval is based on.

Hagelbäck and Johansson [22] tested run-time dynamic difficulty scaling in a real-time strategy game and discovered that players, on the whole, reacted positively to run-time adaptation, but that transparency of the mechanism was a major moderating variable—one of the earliest empirical findings directly inspiring the present work's focus on visible phase transitions. Silva, Silva and Chaimowicz [23] extended DDA for MOBA games, underlining the complexity of multi-player settings where individual performance metrics are influenced by team dynamics.

Rosa et al. [24] presents a DDA approach for platform games using player profiles based on performance and affective information, showing that hybrid profile-based adaptation is superior to only performance-based adaptation. Their work confirms the usefulness of the K-means player profiling module in the present framework. Zook and Riedl [25] developed a temporal data-driven player model for DDA, demonstrating that performance history leads to better adaptation than instantaneous metrics—providing motivation for the exponential smoothing component in this system.

### 2.4. Machine Learning, Fuzzy Logic and RL-Based DDA

The machine learning models for DDA include supervised, unsupervised and reinforcement DDA models. Or, Kolomenkin, and Shabat [8] introduce DL-DDA, a deep neural network that is trained on player telemetry data to predict the best difficulty adjustments within gameplay constraints. While DL-DDA obtains strong benchmark results, it needs a large amount of inference compute, making it less applicable for new or small-scale games. Romero-Mendez et al. [6] also employ a feedforward neural network to pre-

dict player skill in a Space Invaders-like arcade, which then activates predefined difficulty executions; their evaluation with 155 participants reported 90% game enjoyment, serving as a useful benchmark for the current study's 83.3% enjoyment outcome.

Paraschos and Koulouriotis [26] introduce fuzzy-logic focused DDA controllers as an alternative to the threshold-based systems, stating that fuzzy limits are a better representation of the gradual nature of player skill and difficulty. Damastuti et al. [27] integrate Q-learning and fuzzy logic for dynamic level adjustment and show that their proposal is more stable compared to pure rule-based approaches. Fuchs, Gieseke and Dockhorn [28] introduce a personalized DDA system based on imitation learning and reinforcement learning and demonstrate that RL enhancement of rule-based priors leads to faster adaptation convergence—a concept particularly pertaining to future work of the proposed framework. Guo, Thawonmas and Ren [7] challenged the notion that traditional DDA formulations are adequate for today's game design and call for richer player models as well as more nuanced adaptation targets that are not restricted to scalar difficulty parameters.

Mortazavi, Moradi, and Vahabie [5] summarized DDA techniques, reporting that most of the studies provide positive results on engagement, flow, motivation and immersion but there were still gaps in terms of generalizability, transparency and engine independence—exactly what this work aims to close. Exploring DDA for FPS games based on performance and physiological sensors, Fisher and Kulshreshtha [29] revealed that multi-modal adaptation leads to better performance compared with performance-based approaches, indicating a potential direction for this work.

### 2.5. Player Profiling and Behavioral Telemetry

Player profiling—the computational identification of different player types based on their behavioral data—is a prerequisite for personalized DDA. Bauckhage, Drachen, and Sifa [30] offer a tutorial and survey on clustering game behavior data, position K-means at the core of behavioral segmentation, and address pitfalls such as the validity of small clusters and the need for feature standardization. Drachen, Canossa, and Yannakakis [31] applied player profiling to the game Tomb Raider: Underworld with behavioral telemetry to show that unsupervised clustering of players according to in-game behavioral metrics results in interpretable and actionable player segments, offering direct validation of the approach in the present work.

Sifa et al. [32] extends player profiling to Steam platform data spanning millions of players and thousands of games with findings of stable behavioral archetypes that generalize across titles. Their results provide indications that the four profiles extracted in this study (Reckless, Struggling, Defensive and Dominant) are not merely Skyrim-specific phenomena but possibly universal engagement archetypes. Swanson et al. [33] also employed K-means clustering on telemetry of an educational game and found silhouette analysis better than WCSS alone for choosing a robust number of clusters. This was one of the inspirations for including silhouette score validation in selection of K in the present work. Harpstead et al. [34] derived engagement profiles from game log data and demonstrated that behavioral segmentation can capture not only skill level but qualitative differences in engagement style which is the Dominant vs. Defensive distinction in the present results.

### 2.6. Adaptation Transparency and Player Perception

An as-yet-unexplored facet of DDA research is whether players should be able to perceive or know about the adaptation. Hunicke and Chapman [17] argued that invisible adjustment could be more acceptable since players own the outcome of their skills, rather than being assisted by the system. However, Andrade et al. [19] observed that the percep-

tion of fairness increased with adaptation regardless of transparency in the context of an RPG. This work argues that explicit, action-based DDA—where phase shifts are explicitly represented as boss behavior changes—is hypothesized to be preferable to implicit statistical scaling: it maintains player agency and therefore provides a closer perceived linkage between performance and boss response. The 93.3% behavioral awareness result confirms the adaptation was in fact perceptible, which is a necessary precondition for this design argument, though the comparison itself remains untested.

### 2.7. Comparative Summary and Research Gap

Table 1 summarizes key features of ten representative systems related to the proposed framework. Across all the reviewed systems, no prior work has combined behavioral adaptation (rather than statistical) with engine independence, explicit player profiling with clustering and empirical evaluation of perceived-player transparency. The present work targets all four gaps simultaneously.

**Table 1.** Comparative feature analysis of DDA systems.

| System                     | Adaptation Target                         | Mechanism                | Real-Time          | Player Transparency      | Human Evaluation | Key Limitation                          |
|----------------------------|-------------------------------------------|--------------------------|--------------------|--------------------------|------------------|-----------------------------------------|
| RE4 AGS [16]               | Statistical                               | Encounter-based          | Yes                | Low                      | No               | Engine-coupled and invisible scaling    |
| L4D AI Director [18]       | Spawn rate                                | Pacing-based             | Yes                | Low                      | No               | Engine-coupled. Co-op only              |
| Andrade et al. [19]        | Statistical                               | Rule-based               | Between session    | Low                      | Yes              | Statistical and only between sessions   |
| DL-DDA [8]                 | Statistical                               | Neural network           | Yes                | Low                      | Benchmark        | Requires large dataset                  |
| Spronck et al. [14,15]     | Rule weights                              | Dynamic scripting (RL)   | Between encounters | Not reported             | Yes              | Requires predefined rule/tactic library |
| Hagelbäck & Johansson [22] | Statistical                               | Run-time scaling         | Yes                | Variable (tested factor) | Yes              | Single RTS game; mechanism not detailed |
| Rosa et al. [24]           | Profile-based (performance + affect)      | Hybrid profiling         | Not reported       | Not reported             | Not reported     | Requires affective-state sensing input  |
| Damastuti et al. [27]      | Statistical                               | Q-learning + fuzzy logic | Yes                | Not reported             | Not reported     | Evaluated via simulation only           |
| Fisher & Kulshreshtha [29] | Multi-modal (performance + physiological) | Sensor-fusion-based      | Yes                | Not reported             | Yes              | Requires physiological sensor hardware  |
| This Work                  | Behavior                                  | Phase-based behavioral   | Yes (2 s)          | High                     | Yes ( $n = 30$ ) | One-way and single boss                 |

## 3. Materials and Methods

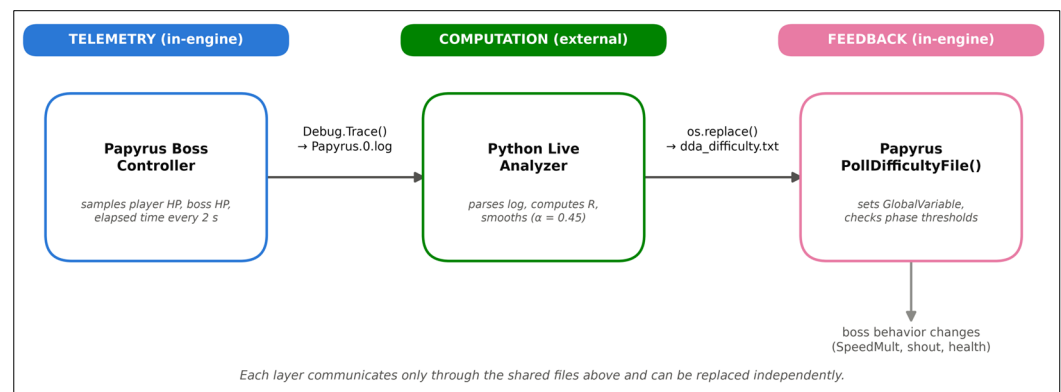
### 3.1. System Architecture

The framework consists of two modules communicating through a local file system, the Papyrus Boss Controller (in-engine) and the Python Live Analyzer (external). During the boss fight, the Papyrus controller samples player health, boss health, and elapsed time

at 2 s intervals and writes structured telemetry entries to the Papyrus debug log through `Debug.Trace()`. The Python analyzer monitors this log file live by parsing entries, computing a difficulty multiplier and writing the results atomically to the shared text file in the SKSE plugins directory. On every `OnUpdate` event, the Papyrus script polls the file and updates a global difficulty variable. When a specific threshold is crossed, the boss transitions to the next behavioral phase.

The file-based communication was chosen for simplicity and compatibility with Skyrim's scripting environment avoiding any direct sockets or REST endpoints that would need engine-level modifications. The system comprises three independent layers: Telemetry (Papyrus log writing), Computation (external Python difficulty calculation) and Feedback (global variable update and phase transition execution). Each layer can be replaced or modified without affecting the others.

We trace one complete update cycle through Figure 1: the Papyrus Boss Controller samples player HP, boss HP and elapsed time every 2 s and writes each sample via `Debug.Trace()` into `Papyrus.0.log`. The Python Live Analyzer, running externally, tails this log, parses each new sample, computes the dominance ratio  $R$ , and applies exponential smoothing to obtain the new difficulty value. This value is then written atomically to `dda_difficulty.txt` using `os.replace()`, preventing Papyrus from ever reading a partially written file. On its next `OnUpdate` event, the Papyrus Boss Controller polls this file via `PollDifficultyFile()`, updates the corresponding `GlobalVariable`, and checks it against the phase thresholds; crossing a threshold triggers the corresponding behavioral change (Sprint Charge, Unrelenting Force shout, or the health/speed boost). This full loop—from a single 2 s sample to a possible visible change in boss behavior—repeats continuously for the duration of the fight, with the 10 s cooldown between difficulty updates.



**Figure 1.** System architecture diagram: data flow through the three-layer pipeline.

### 3.2. Difficulty Computation Model

The difficulty computation pipeline is formalized in Algorithm 1. Raw health values are sampled at 2 s intervals. Player damage taken per second ( $D_p$ ) and damage dealt to the boss per second ( $D_n$ ) are derived from health deltas with  $\max(0, \Delta\text{Health})$  clamping to exclude player healing and boss health modifications from phase transitions. A minimum of four samples and a four-second window are required before computation begins to prevent premature difficulty decisions from statistically insufficient data.

The dominance ratio  $R$  is computed as

$$R = D_n / (D_n + D_p), \quad (1)$$

where  $D_n$  is the rate at which player deals damage to the boss and  $D_p$  is the rate at which the boss deals damage to the player. As  $R$  approaches 1.0, the player dominates and as  $R$

approaches 0, the player struggles. Note that  $D_n$  and  $D_p$  are non-negative after clamping. The same naming convention is used throughout the whole paper.

The ratio is linearly mapped to a target difficulty in  $[D_{\min}, D_{\max}]$  as  $[1.0, 1.35]$ . Each update increments the difficulty value by a fixed step size of 0.07 applied to the computed target to ensure gradual escalation rather than abrupt spikes. A dominance boost multiplier of  $1.6\times$  is applied when the target exceeds current difficulty by more than 0.10, accelerating escalation for dominant players. The raw update is applied with an exponential smoothing as

$$D_{\text{new}} = D_{\text{current}} + 0.45 \times (D_{\text{raw}} - D_{\text{current}}), \quad (2)$$

included with a 10 s cooldown between updates to prevent any erratic oscillations from momentary spikes of performance. The parameter choices were made as follows:  $\alpha = 0.45$  was chosen as a moderate smoothing factor that responds to trends within the 3–4 update cycles while removing single cycle outliers. Lower values such as 0.2 produced under-responsive trajectories in informal tuning during development and high values above 0.6 caused oscillation.  $D_{\max} = 1.35$  was set in correspondence to Phase 3 difficulty (maximum behavioral state) without exceeding it. The phase thresholds of 1.10, 1.20 and 1.28 were spaced to ensure each phase is triggered within approximately 3–5 update cycles of repetitious dominant play, producing perceptible escalation without immediately overwhelming the player. The 10 s cooldown prevents any consecutive updates within a single combat exchange.

---

#### Algorithm 1. DDA Difficulty Computation Pipeline

---

INPUT: health\_log[] (timestamped player/boss HP samples)

OUTPUT:  $D_{\text{new}}$  (updated difficulty value)

```

1. IF len(health_log) < 4 OR elapsed_time < 4 s THEN
2.     RETURN  $D_{\text{current}}$                                 //insufficient data guard
3. END IF
4.  $D_p \leftarrow \max(0, \text{delta\_player\_HP}/\text{delta\_t})$       //damage taken by player per second
5.  $D_n \leftarrow \max(0, \text{delta\_boss\_HP}/\text{delta\_t})$         //damage dealt to boss per second
6.  $R \leftarrow D_n/(D_n + D_p + \epsilon)$                   //dominance ratio ( $\epsilon$  avoids/0)
7.  $D_{\text{target}} \leftarrow D_{\min} + R \times (D_{\max} - D_{\min})$     //linear mapping [1.0, 1.35]
8. step  $\leftarrow 0.07$ 
9. IF  $D_{\text{target}} - D_{\text{current}} > 0.10$  THEN
10.    step  $\leftarrow \text{step} \times 1.6$                         //dominance boost
11. END IF
12.  $D_{\text{raw}} \leftarrow D_{\text{current}} + \text{step} \times \text{sign}(D_{\text{target}} - D_{\text{current}})$ 
13.  $D_{\text{new}} \leftarrow D_{\text{current}} + 0.45 \times (D_{\text{raw}} - D_{\text{current}})$  //exponential smoothing  $\alpha = 0.45$ 
14. IF elapsed_since_last_update < 10s THEN
15.    RETURN  $D_{\text{current}}$                                 //cooldown
16. END IF
17. WRITE  $D_{\text{new}}$  atomically to shared file (os.replace())
18. RETURN  $D_{\text{new}}$ 

```

---

### 3.3. Phase Based Behavioral Adaptation

The boss functions across four states. Phase 0 ( $D < 1.10$ ) is the baseline Skyrim AI with no DDA modifications. Phase 1 ( $D \geq 1.10$ ) unlocks a Sprint Charge ability by increasing the boss's SpeedMult by +50 and forcing AI re-evaluation through EvaluatePackage(). Phase 2 ( $D \geq 1.20$ ) adds an Unrelenting Force shout via AddShout() which immediately ragdolls the player on phase entry, sets boss aggression to 3 and initiates periodic 12 s shout cycles (does not ragdoll the player this time). Phase 3 ( $D \geq 1.28$ ) applies a modification of +200 to the base health via ModActorValue(), increases SpeedMult further by +15

and continues periodic shouts. The transitions are unidirectional by design with no de-escalation.

Atomic file writing was implemented through the `os.replace()` pattern in Python that guarantees that Papyrus never reads any partially written difficulty value during a concurrent write. A false-trigger prevention mechanism is included in `EndCombat()` using a `Utility.Wait(1.5)` followed by `Self.IsInCombat()` for reverification to differentiate genuine combat end events from `StopCombat()` calls triggered internally by the Unrelenting Force shout animation.

### 3.4. Implementation Environment

The Papyrus Boss Controller was implemented as a script extending the Actor class in Bethesda's Creation Kit [35] (v1.6.1378.1) attached to a custom Draugr Overlord NPC. The Python live analyzer was implemented in Python 3.11.1 using standard library modules (`re`, `os`, `csv`, `time`). Scikit-learn [36] (v1.9.0) was used for post-test clustering analysis. PapyrusUtil [37] (Version 4.6), an SKSE plugin, provided `MiscUtil.ReadFile()` for in-game file polling. The game client was Skyrim Special Edition [38] (Version 1.6.1170) with SKSE [39] (Version 2.2.6) installed alongside.

### 3.5. Experimental Design and Participants

Thirty volunteer participants across various skill levels (33.3% beginner, 20.0% casual, 23.3% experienced, 23.3% veteran;  $n = 10, 6, 7, 7$  respectively) completed multiple boss fight sessions (166 total runs). Four participants each left exactly one item unanswered: three (Tester 16: "deserved rather than unfair"; Tester 19: "behavior changes feel connected to how well you were doing"; Tester 22: "boss difficulty felt appropriate for my skill level") each skipped a different Likert item, each reported at  $n = 29$  in Table 2; the fourth (Tester 10) skipped only the self-reported session-count question, which is not part of Table 2 and is superseded by the telemetry-derived count in Section 4.1. All other items and all telemetry-based counts are at  $n = 30$ . To control equipment and character builds, all participants used a standardized character—level 20, Steel Warhammer, Steel armor set, five healing potions and five stamina potions. The sessions were conducted in a controlled indoor vanilla Skyrim cell to eliminate any environmental variables (terrain, lighting, NPCs). The authors were present during all sessions to confirm telemetry logging and resolve any setup issues.

The post-test data collection used two tools: (1) Google Form using Likert-scale questions assessing engagement, perceived fairness, difficulty naturalness and progression, behavioral awareness and overall player enjoyment and (2) per-session telemetry CSV logs recording fight duration,  $D_p$ ,  $D_r$ , kill/death outcomes, final HP and maximum difficulty reached. Kill rate and death rate are defined as the proportion of sessions per participant ending in a boss kill or a player death, respectively. The remaining sessions ended without a decisive outcome (including fights the participant abandoned) and are excluded from both rates.

The questionnaire items were adapted from constructs in the games-user-research literature rather than administered as a validated psychometric scale. Engagement and re-engagement items were inspired by flow theory [2,3] and the immersion criteria by the GameFlow model [10]; perceived fairness and challenge-skill balance items draw on the constructs measured by Andrade et al. [19] and on the challenge-skill balance, which are central to flow theory [2,3]; the control item draws on the GameFlow control criterion [10]; the single enjoyment item follows the GameFlow enjoyment criterion [10]; and the two behavioral-awareness items (noticing behavioral change, and perceiving that change as connected to performance) constitute a novel construct introduced by this study rather than one drawn from prior instruments. The comparative item ("the fight held my atten-

tion better than a typical Skyrim boss”) is treated as a standalone measure given its explicit static-difficulty referent and is not pooled into any composite score. Internal consistency (Cronbach’s  $\alpha$ ) was estimated for each multi-item construct group: perceived fairness/challenge-skill balance (4 items) showed acceptable consistency ( $\alpha = 0.70$ , 95% CI [0.47, 0.85],  $n = 28$ ); engagement (2 items,  $\alpha = 0.04$ ) and behavioral awareness (2 items,  $\alpha = 0.15$ ) showed low consistency, consistent with two-item subscales yielding conservative, unstable estimates at this sample size. Given these values, engagement and behavioral awareness are reported at the item level (Table 2) rather than as composite scale scores.

The study was conducted in accordance with the Declaration of Helsinki. Informed consent was obtained from participants prior to participation. All participant data were anonymized, and testers are identified by number ID only throughout this paper.

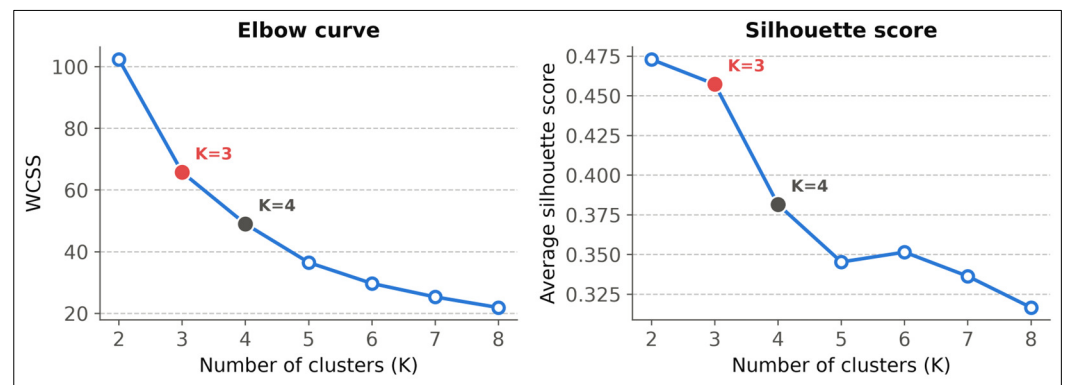
**Table 2.** Likert-scale survey results.

| Survey Question                             | 1 | 2 | 3  | 4  | 5  | Mean        | SD   | Median | 95% CI          |
|---------------------------------------------|---|---|----|----|----|-------------|------|--------|-----------------|
| Fight was engaging                          | 0 | 0 | 1  | 11 | 18 | <b>4.57</b> | 0.57 | 5      | [4.36, 4.77]    |
| Motivated to re-engage after death          | 2 | 1 | 1  | 9  | 17 | 4.27        | 1.14 | 5      | [3.86, 4.68]    |
| More engaging than static Skyrim boss       | 0 | 2 | 13 | 10 | 5  | 3.60        | 0.86 | 3.5    | [3.29, 3.91]    |
| Difficulty escalation felt deserved *       | 1 | 3 | 4  | 8  | 13 | 4.00        | 1.16 | 4      | [3.58, 4.42]    |
| Difficulty progression felt natural         | 1 | 3 | 4  | 13 | 9  | 3.87        | 1.07 | 4      | [3.48, 4.25]    |
| Difficulty felt well-calibrated *           | 2 | 7 | 6  | 8  | 6  | <b>3.31</b> | 1.26 | 3      | [2.85, 3.77]    |
| Boss felt harder during session (Yes/No) ** | — | — | —  | —  | —  | 93.3%       | —    | —      | [78.7, 98.2%] * |
| Noticed behavioral changes in boss          | 0 | 2 | 1  | 12 | 15 | 4.33        | 0.84 | 4.5    | [4.03, 4.64]    |
| Boss behavior connected to my performance * | 1 | 2 | 3  | 16 | 7  | 3.90        | 0.98 | 4      | [3.54, 4.25]    |
| Overall enjoyment                           | 0 | 3 | 2  | 8  | 17 | 4.30        | 0.99 | 5      | [3.95, 4.65]    |
| Challenge felt well-balanced                | 1 | 3 | 7  | 11 | 8  | 3.73        | 1.08 | 4      | [3.35, 4.12]    |
| Felt in control of the outcome              | 0 | 5 | 7  | 11 | 7  | 3.67        | 1.03 | 4      | [3.30, 4.03]    |

**Notes:** The scale is as follows: 1 = Strongly Disagree, and 5 = Strongly Agree. \* Each starred item is missing a response from one respondent; a different single respondent is missing for each of the three starred items ( $n = 29$  for that item; see Section 3.5 for which respondent). \*\* Binary Yes/No item;  $n = 30$ ; confidence interval computed via the Wilson score method (appropriate for a proportion this close to 1). Bold indicates the highest and lowest mean ratings among the eleven Likert-scale items.

### 3.6. Post Test Analysis: K-Means Clustering and t-SNE Visualization

K-means clustering was applied to a  $30 \times 7$  feature matrix of the aggregated per-tester telemetry from average fight duration, average  $D_p$ , average  $D_n$ , average max difficulty reached, kill rate, death rate and damage efficiency ratio ( $D_n/D_p$ ). All features were standardized for z-score prior to clustering. The optimal number of clusters  $K$  was selected using the standard elbow method, within-cluster sum of squares (WCSS) plotted for  $K = 2$  through  $K = 8$  (Figure 2) and the average silhouette score computed for each  $K$  using scikit-learn’s silhouette score function [36,40]. The silhouette score  $s(i)$  for each data point  $i$  is defined as  $s(i) = (b(i) - a(i)) / \max\{a(i), b(i)\}$ , where  $a(i)$  is the mean intra-cluster distance and  $b(i)$  is the mean distance to the nearest other cluster. The scores range from  $-1$  to  $1$  with higher values indicating better-defined clusters. The t-SNE dimensionality reduction ( $7D \rightarrow 2D$ , perplexity = 8, 2000 iterations) was applied to the same feature matrix to produce an illustrative projection of the cluster structure [41]. The t-SNE is used for visualization purposes only and not as a validation metric. All stochastic procedures (K-means initialization, t-SNE) used a fixed random seed (42) for reproducibility; analysis scripts are available alongside the data cited in the Data Availability Statement.



**Figure 2.** Elbow curve and silhouette score by number of clusters (K).

## 4. Results

### 4.1. Survey Results

A total of 100% (30/30) of participants completed four or more fight sessions, recomputed directly from telemetry rather than self-report (every participant logged at least five sessions; range 5–9) and 86.7% (26/30) reported perceiving variance between attempts confirming that the DDA framework maintained meaningful differences between runs. Table 2 presents the full Likert distribution, mean, standard deviation, median and 95% confidence interval for each Likert-scale item.

#### 4.1.1. Player Engagement and Flow

Player engagement was the highest result in the survey, with a combined 96.7% (29/30) of participants rating the fight as engaging (Likert 4 or 5) with 60.0% (18/30) at the maximum score (mean = 4.57 and SD = 0.57). The post-death motivation to re-engage was a direct indication of flow state maintenance which yielded an 86.7% (26/30) positive response (Likert 4 or 5). Because no static control condition was used, testers' comparison with a typical static Skyrim boss encounter is only indicative: 50.0% (15/30) rated the adaptive boss higher in sustained attention (Likert 4 or 5).

#### 4.1.2. Perceived Fairness and Difficulty Escalation Naturalness

Participants were asked whether difficulty escalation felt deserved rather than unfair and 72.4% (21/29) agreed (Likert 4 or 5) with 44.8% (13/29) at maximum score (mean = 4.00, SD = 1.16). The natural and gradual difficulty progression received 73.3% (22/30) agreement (Likert 4 or 5, mean = 3.87, SD = 1.07), supporting that the exponential smoothing successfully prevented any jarring difficulty spikes. The perceived appropriateness of difficulty yielded a combined 48.3% (14/29) at Likert 4 or 5.

#### 4.1.3. Behavioral Awareness

In terms of observing the behavioral changes in the boss, all results were positive. A binary question was asked whether the boss felt harder during the session and was positively answered by 93.3% (28/30) of the participants. Awareness of the boss's behavioral changes also received strong agreement, achieving 90.0% (27/30; mean = 4.33, SD = 0.84) at Likert 4 or 5. Participants identified specific phase behaviors by 86.7% (shout), 73.3% (speed boost) and 63.3% (boss power-up and health increase) directly corresponding to Phase 2, Phase 1 and Phase 3 respectively. The perceived connection between player performance and boss behavior received an agreement of 79.3% (23/29) at Likert 4 or 5 (mean = 3.90, SD = 0.98).

#### 4.1.4. Overall Enjoyment and Challenge Balance

Overall player enjoyment was high with 83.3% (25/30) and no participants rating it at 1 (mean = 4.30, SD = 0.99). The challenge balance (Likert 4 or 5) yielded 63.3% (19/30) agreement and perceived outcome control reached 60.0% (18/30).

#### 4.1.5. Exploratory Inferential Statistics

As no hypotheses were pre-registered, the following tests are exploratory. One-sample Wilcoxon signed-rank tests were run for each Likert item against the neutral midpoint of 3, with Holm correction across the eleven tests and effect size  $r = |Z|/\sqrt{N}$ . Ten of eleven items differed significantly from neutral after correction (all above it), including engagement (median 5, Holm  $p < 0.001$ ,  $r = 0.89$ ), overall enjoyment (median 5, Holm  $p < 0.001$ ,  $r = 0.79$ ) and behavioral awareness (median 4.5, Holm  $p < 0.001$ ,  $r = 0.82$ ). The exception was “the boss difficulty felt appropriate for my skill level” (Table 2: “Difficulty felt well-calibrated”; median 3, Holm  $p = 0.18$ ,  $r = 0.25$ ), a response statistically indistinguishable from neutral. The comparative item “the fight held my attention better than a typical Skyrim boss” (Table 2: “More engaging than static Skyrim boss”) also exceeded the midpoint (median 3.5, Holm  $p = 0.007$ ,  $r = 0.57$ ); however, its modal response was neutral (13/30, 43%), and the test’s significance depends on discarding these neutral responses before ranking. As a single retrospective self-report item, it does not constitute a controlled comparison against static difficulty and is not interpreted as evidence that the adaptive system is more engaging than a static one; that comparison requires the controlled study identified in the Limitations and Future Work sections.

#### 4.1.6. Volunteered Comments

Two of the questionnaire’s text fields were optional. In addition, 26 of 30 participants left a substantive comment in at least one. These are anecdotal volunteered comments, not a structured qualitative analysis, and are reported here to include critical as well as positive reactions rather than only the latter. Comments were predominantly positive and frequently referenced the adaptive behavior specifically rather than the encounter in general—e.g., Tester 17: “Impressive system. I could feel the boss adapting in real time, great work”; Tester 19: “Solid mechanics. The difficulty felt earned rather than artificial.” One participant raised a clear criticism of the escalation pace: Tester 13 wrote, “The power creep is way too sharp. When you do relatively decent, the boss will be way too tough the subsequent runs.” This concern is consistent with the existence of a profile—Struggling—for which escalation appears to outpace recovery (56% death rate, Table 3), and suggests the escalation rate may feel disproportionate to players who begin performing well but cannot sustain it—a candidate direction for the planned follow-up study’s structured interviews.

The rationale for  $K = 4$  cluster selection and the description for each profile are explained below, in Section 4.2.

**Table 3.** K-means cluster centroids ( $K = 4$ ; mean feature values per player profile,  $n = 30$ ).

| Profile    | <i>n</i> | Avg Duration (s) | Kill Rate   | Death Rate  | Efficiency  | Avg D <sub>p</sub> (DPS Taken) | Avg Max Diff. | Silhouette |
|------------|----------|------------------|-------------|-------------|-------------|--------------------------------|---------------|------------|
| Reckless   | 3        | ~39              | 0.00        | 0.09        | 0.87        | <b>4.18</b>                    | 1.14          | 0.77       |
| Struggling | 4        | ~77              | 0.00        | <b>0.56</b> | 0.82        | 1.52                           | 1.20          | 0.39       |
| Defensive  | 7        | ~163             | 0.21        | 0.13        | 0.73        | 0.63                           | 1.27          | 0.34       |
| Dominant   | 16       | ~291             | <b>0.36</b> | 0.03        | <b>2.65</b> | 0.22                           | <b>1.33</b>   | 0.32       |

**Notes:** Bold indicates the highest value among the four profiles for each behavioral metric (kill rate, death rate, efficiency,  $D_p$ , and max difficulty). Kill rate and death rate are defined in Section 3.5.

#### 4.2. Cluster Count Selection and K-Means Results

The within-cluster sum of squares (WCSS) decreases smoothly across  $K = 2$ –8 with no sharp inflection (Figure 2): proportional reductions of 36%, 25%, 26%, 19%, 15% and 14% for each successive  $K$ . The  $K = 3 \rightarrow 4$  and  $K = 4 \rightarrow 5$  reductions are nearly identical (25% and 26%), so the elbow curve is flat through  $K = 4$  rather than bending at it, and the elbow method alone does not discriminate among  $K = 3$ ,  $K = 4$ , or neighboring values.

The average silhouette score was computed for  $K = 2$  through  $K = 8$  on the same standardized  $30 \times 7$  feature matrix (Figure 2, second axis): 0.47, 0.46, 0.38, 0.35, 0.35, 0.34 and 0.32 respectively, peaking at  $K = 2$ .  $K = 2$  is not adopted as the reported solution: it corresponds to a single dominant contrast approximating an overall performance split—a weaker-performance group (combining what the  $K = 4$  solution separately labels Reckless, Struggling, and part of Defensive) against a stronger-performance group (Dominant and the remainder of Defensive)—and collapses the behavioral differentiation this study characterizes. It is also not a robust partition: K-means and Ward hierarchical clustering, both run at  $K = 2$  on the same data, yield only weak agreement (Adjusted Rand Index = 0.34, chance-corrected), indicating the two-way split is orientation-unstable across clustering methods rather than a single well-defined boundary.

Among  $K \geq 3$ ,  $K = 3$  is the statistically preferred solution. It has the second-highest silhouette score (0.46, versus 0.38 at  $K = 4$ ) and, more importantly, substantially higher cluster stability under resampling: an adjusted Rand index of 0.91 across 200 bootstrap replications (80% subsamples), compared with 0.66 at  $K = 4$ —the least stable solution among  $K = 2$  through  $K = 6$  tested. Cross-method agreement tells that K-means and Ward hierarchical clustering agree far more at  $K = 4$  (Adjusted Rand Index = 0.89) than at  $K = 3$  (0.57) or  $K = 2$  (0.34), meaning the four-group partition itself is not an artifact of the K-means algorithm specifically, even though it is less stable than  $K = 3$ . The three  $K = 3$  profiles are:

1. Reckless ( $n = 3$ )—Average duration ~39 s,  $D_p$  4.18,  $D_n$  3.61, 0% kill rate, 9% death rate, efficiency 0.87.
2. Contested ( $n = 8$ )—Average duration ~104 s,  $D_p$  1.16,  $D_n$  0.79, 7% kill rate, 35% death rate, efficiency 0.73.
3. Dominant ( $n = 19$ )—Average duration ~278 s,  $D_p$  0.26,  $D_n$  0.49, 36% kill rate, 4% death rate, efficiency 2.37.

Reckless is the most genuinely distinct group in the data (per-cluster silhouette 0.77, essentially unchanged across every  $K$  from 2 to 8) and is never merged with any other group at any  $K$ , including at  $K = 3$ .

Increasing to  $K = 4$  breaks the  $K = 3$  Contested cluster: four of its eight members retain short fight durations and elevated death rates, forming the Struggling profile ( $n = 4$ ), while the other four—whose fight durations and difficulty trajectories more closely resemble the Dominant group—form the Defensive profile ( $n = 7$ ) alongside a matching subset drawn from Dominant. This split is close to even (4 of 8), consistent with the fourth cluster subdividing a continuum within Contested rather than separating out a structurally distinct class.

The  $K = 4$  solution is retained in the profile descriptions and Table 3 above for interpretability and design relevance: the Defensive/Struggling distinction differentiates two behaviorally and DDA-relevant patterns—cautious, moderate-length engagement versus rapid, repeated failure—that the coarser  $K = 3$  Contested cluster does not distinguish. This

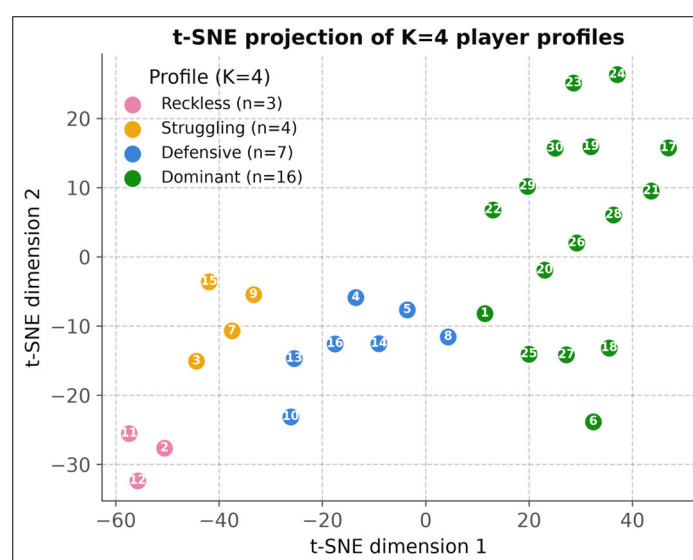
is an interpretability-motivated choice, not a statistically superior one.  $K = 4$ 's silhouette (0.38) and bootstrap stability (0.66) are both lower than  $K = 3$ 's, and any claim specific to the Defensive profile is read with this in mind: of the  $K$  values tested,  $K = 4$  is the least robust to which participants happen to be sampled, and the Defensive/Struggling boundary in particular is descriptive rather than evidence of two freestanding, reproducible player types.

The four  $K = 4$  profiles (Table 3) are:

4. Reckless ( $n = 3$ )—Shortest fight duration ( $\sim 39$  s) and the highest  $D_p$  (4.18), indicating high-risk play with limited defensive awareness. 0% kill rate, 9% death rate; the remaining 91% of sessions ended without a decisive outcome (neither a boss kill nor a player death).
5. Struggling ( $n = 4$ )—Highest death rate (56%) and 0% kill rate; the DDA rarely escalated to the highest phase for this group and difficulty remained comparatively low.
6. Defensive ( $n = 7$ )—Lower  $D_p$  (0.63) than Reckless or Struggling and a 21% kill rate; the DDA escalated to Phase 1 and Phase 2 for this group.
7. Dominant ( $n = 16$ )—The largest group, longest average fight duration ( $\sim 291$  s), lowest  $D_p$  (0.22), highest kill rate (36%) and highest damage efficiency ratio (2.65); the DDA consistently drove to the maximum difficulty for this group.

#### 4.3. t-SNE Cluster Visualization

The t-SNE projection used for visualization (perplexity = 8, 2000 iterations) shows the clear spatial separation between the four K-means clusters in 2D embedding (Figure 3). The Dominant group occupied the upper-right region, Reckless was isolated and plotted on the lower-left, Struggling appeared in the middle-left and Defensive occupied most of the central region. This spatial separation helps us interpret the point groupings as the four profiles, but it does not validate the cluster count itself. Cluster-count validity is assessed via the silhouette and stability analysis in Section 4.2, which favors  $K = 3$  statistically;  $K = 4$  is retained here for interpretability despite that evidence, not because of it.



**Figure 3.** t-SNE 2D projection of the 7-feature player telemetry space, colored by K-means cluster (perplexity = 8,  $n = 30$ ).

#### 4.4. Individual Trace Analysis—Tester 1

An individual trace analysis was conducted, and Tester 1 was picked as their seven-session history showed a clear learning arc that provides the best narrative for the DDA

analysis. Tester 1 was classified in the Dominant cluster with an average fight duration of 195.7s, kill rate of 42.9%, damage efficiency of 1.67 and average max difficulty reached at 1.31. They were placed at the lower boundary of the Dominant cluster. Table 4 represents their seven-session DDA trace. Some negative DPS values in sessions 2–7 are Papyrus logging artifacts: negative  $D_p$  values most likely reflect player-healing events between samples, while negative  $D_n$  values stem from Phase 3 boss health modifications; both were clipped to zero for analysis purposes.

**Table 4.** Tester 1—DDA Trace.

| Session | Duration (s) | $D_p$   | $D_n$   | Max Diff. | Final HP | Phase   | Outcome | DDA Decision          |
|---------|--------------|---------|---------|-----------|----------|---------|---------|-----------------------|
| 1       | 32           | 1.61    | 2.78    | 1.183     | 10.6     | Phase 1 | Death   | Correct: limited data |
| 2       | 106          | −0.10 * | 0.21    | 1.313     | 24.0     | Phase 3 | Death   | Correct: dominant     |
| 3       | 84           | 0.62    | −1.34 * | 1.313     | 29.7     | Phase 3 | Death   | Correct: dominant     |
| 4       | 188          | 0.15    | 0.35    | 1.318     | 31.8     | Phase 3 | Death   | Correct: dominant     |
| 5       | 328          | −0.05 * | 0.14    | 1.327     | 45.4     | Phase 3 | Kill    | Correct: dominant     |
| 6       | 190          | −0.05 * | 0.46    | 1.350     | 51.7     | Phase 3 | Kill    | Correct: dominant     |
| 7       | 442          | −0.03 * | 0.10    | 1.350     | 123.8    | Phase 3 | Kill    | Correct: dominant     |

**Notes:**  $D_p$  = player DPS taken;  $D_n$  = boss DPS dealt by player. \* Represents a Papyrus logging artifact and is clipped to 0 for analysis (see Section 4.4).

In Session 1 (32s), the DDA never increased beyond Phase 1 because there was insufficient data, and the performance signal was too vague. Sessions 2–4 reached Phase 3, as expected from a Dominant profile player. The deaths show a learning curve. Sessions 5–7 produced boss kills at max difficulty, with final player HP increasing from 45.4 to 123.8 indicating player skill growth as predicted by flow theory’s growing competence under sustained calibrated challenge [2,3].

#### 4.5. Controller Statistics Across Sessions

Because phase transitions are unidirectional (no de-escalation), the maximum difficulty reached during a fight directly indicates the highest behavioral phase triggered in that session. Across all 166 sessions, 93.4% reached at least Phase 1 (Sprint Charge), 77.1% reached at least Phase 2 (Unrelenting Force shout), and 62.7% reached Phase 3 (health and speed boost). Phase-reach rates differed substantially by  $K = 4$  profile: Dominant players reached Phase 3 in 96.3% of their sessions, Defensive players in 54.1%, Struggling players in 19.2%, and Reckless players in none of their sessions (0%), consistent with their ~39 s average fight duration leaving insufficient time for escalation. This confirms the controller’s phase transitions responded differentially to player performance across the identified profiles.

Two further controller metrics—the number of difficulty updates per session and the rate at which the 10 s cooldown blocked an update—would require the per-2 s telemetry sample stream underlying each fight. Only session-level aggregates (final HP values, duration, and maximum difficulty reached) were retained in the available data, so these two metrics could not be computed; see Limitations.

## 5. Discussion

### 5.1. Principal Findings

The results demonstrated that the modular engine-independent DDA with behavioral changes is practically achievable and positively received by players across various skill levels. The framework successfully addressed all four gaps identified in the literature review—engine independence, behavioral (rather than statistics-based) adaptation, ex-

plicit player profiling, and empirical evaluation of perceived transparency. A total of 93.3% of participants answered yes to the binary awareness question, and the Likert behavioral-awareness item (90.0%) was among the highest-agreement items, indicating that observable phase-based adaptation is not only perceptible but also interpreted by most players as a meaningful response to their battle performance.

The K-means clustering identified four clear profiles within the expectations of the difficulty model. The Dominant cluster's prevalence of 53.3% ( $n = 16$ ) suggests that  $D_{\max} = 1.35$  may be insufficient for higher skilled players which represents a limitation in calibration for experienced audiences. Future calibration would extend further and add a Phase 4 to serve this population. This finding is consistent with Rosa et al. [24] who found that profile-based adaptation must account for the upper level of the skill distribution.

The individual trace analysis for Tester 1 illustrates that the DDA made contextually appropriate decisions across the seven sessions. In Session 1, limited data led to limited escalation, which was the right decision, preventing an unfair early difficulty spike. Sessions 5–7 showed a strong learning arc with increasing final HP, consistent with Csikszentmihalyi's flow theory [2,3].

Andrade et al. [19], the closest academic precedent, reported statistically significant improvements in enjoyment and fairness. The survey responses of this study point in the same direction, but were collected without a control condition. This means we cannot interpret them as confirming the improvements of Andrade et al., so establishing if the proposed framework offers a measurable improvement over static difficulty will be the primary direction for future work.

## 5.2. Limitations

The most important limitation is the absence of a static difficulty control condition for comparison. It is clear that the DDA system worked as intended and was well received. However, in the absence of a control group, it is not possible to establish whether it is better or worse than static difficulty, especially since all participants played only the adaptive version—we cannot rule out that part of the positive responses came from the novelty of playing a modded, adaptive encounter for the first time, rather than from the DDA mechanism itself. Future work will address this limitation by testing against a control group.

Two of the four clusters—Reckless and Struggling—have only three and four participants, so their centroids should be interpreted as descriptive rather than as statistical profiles that would necessarily reappear in larger samples (see Section 4.2 for the full cluster-stability analysis).

The framework was tested on a single boss, inside one specific game, and with a small sample size of only 30 participants. While the architecture of the framework is designed to be general and easy to use in any game, the current study has not demonstrated this aspect yet. Section 5.4 discusses this in more detail—in short, the specific parameter values and player profiles reported here should be read as a case study, not a validated general-purpose calibration.

The statistical tests reported in Section 4.1.5 are exploratory. We did not pre-register any hypotheses, and the eleven tests were run on the same data used to describe the survey results in the first place. Holm correction reduces the risk of false positives across those tests, but it does not make the analysis confirmatory—it should be read as support for the descriptive picture, not as independent proof of it.

The phase transitions are also one-way by design, so once a player triggers one phase, the boss will never de-escalate even if the player starts struggling later. This was still perceived in a positive light by the players, but it must be acknowledged that testing did not take place in a bidirectional DDA context and represents a deliberate design trade-off.

Controller validation is also limited to what session-level data allows: we report how often each phase was reached (Section 4.5), but could not test the controller's sensitivity to its own parameters, such as the smoothing factor  $\alpha$ , since only session summaries, not the raw per-sample logs, were available.

### 5.3. Future Work

The future work focuses on six directions. First, a static baseline comparison should be conducted to compare DDA and static difficulty conditions within subjects, to enable causal attribution of engagement effects following Andrade et al. [19].

Second, the difficulty ceiling expansion should be extended beyond Phase 3 by adding Phase 4 behavioral content (additional attack patterns or environmental interactions) for more calibration range for skilled players. Third, bidirectional phase transitions should be implemented through sustained underperformance conditions to allow the system to only de-escalate when a dominant player subsequently struggles. Fourth, reinforcement learning approaches should be investigated by augmenting or replacing the dominance ratio with an RL agent trained to maximize length of session and re-engagement metrics, building on recent advances in DDA personalized through reinforcement learning and imitation [28]. Fifth, multi-session adaptation should be explored by increasing the initial difficulty of each new session based on the player's profile history to reduce the warmup period required to reach the calibrated difficulty. Finally, the framework's architectural generalizability should be evaluated by extending to the Mirmulnir dragon (aerial boss) encounter in Skyrim, which involves flight, positional combat and ranged attacks.

### 5.4. Architectural Scope and Generalizability

The framework's engine independence is an architectural property: the three-layer design communicates only through a shared file interface, so in principle any game or engine capable of writing a text log and reading a text file could support the same telemetry format and computation layer with minimal modification. This is portability in principle. What has actually been demonstrated is narrower: a single boss encounter, in a single game (Skyrim Special Edition), with a standardized character build, evaluated by  $n = 30$  participants. No claim is made that the specific difficulty model, phase thresholds or feature set would transfer unchanged to a different genre, engine or boss design without recalibration. The distinction matters for how the contribution should be read: the modular architecture is the generalizable part; the specific parameter values and player profiles reported here are demonstrated only within this one implementation and should be treated as a case study rather than a validated general-purpose calibration.

### 5.5. Comparison with Existing DDA Approaches

Table 1 (Section 2.7) summarizes ten representative DDA systems; of these, we draw out RE4's Adaptive Game Script, Left 4 Dead's AI Director, DL-DDA and Andrade et al.'s controller for a focused comparison on scalability, modularity and deployment cost specifically, since this bears on the framework's practical contribution beyond behavioral adaptation itself. RE4's Adaptive Game Script and Left 4 Dead's AI Director are both compiled into their respective game engines: adjusting their behavior requires access to and recompilation of the original C++ codebase, which is unavailable in nearly all commercial or modding contexts and makes them non-transferable in practice, whatever their in-game effectiveness. DL-DDA [8] achieves strong benchmark results but depends on a trained deep network that requires substantial telemetry volume and inference compute at runtime, a cost that scales poorly for a small or newly released game without an existing player base to train on. Andrade et al.'s [19] rule-based controller is closer in spirit to the present work and requires comparatively little data, but was implemented and evaluated

only within a custom turn-based prototype rather than a shipped commercial engine. The present framework's file-based interface trades some of DL-DDA's adaptive sophistication for near-zero integration cost: it requires no engine source access, no training data collection phase, and no runtime inference budget beyond parsing a text log and writing a text file every few seconds, at the cost of a simpler rule-based difficulty model than a learned one. This positions the framework as comparatively cheap to deploy and modify, though—consistent with Section 5.4—this deployment-cost advantage has been demonstrated in one integration (Skyrim via Papyrus) rather than validated across multiple engines.

## 6. Conclusions

This paper presented a modular engine-independent DDA framework for boss battles implemented and evaluated in *The Elder Scrolls V: Skyrim*. A dominance ratio difficulty model with exponential smoothing drives the unidirectional phase-based transitions communicating via a shared file interface between the in-engine Papyrus controller and external Python analyzer. The empirical evaluation with 30 participants across 166 sessions showed strong engagement (96.7%), perceived fairness (72.4%) and near-unanimous behavioral awareness (93.3%). These results demonstrate the feasibility and practical applicability of the proposed framework, but they do not establish superiority over static or other implementations of adaptive difficulty, since no controlled comparison was performed. Thus, the results should be interpreted as exploratory in nature, rather than confirmatory. The K-means clustering on seven telemetry features, with cluster count retained for interpretability despite silhouette and stability analysis favoring a coarser solution (Section 4.2), and the t-SNE visualization illustrated four player profiles (Reckless, Struggling, Defensive and Dominant). An individual trace analysis illustrated that the DDA made contextually appropriate decisions across sessions for a representative participant, being consistent with the cluster expectations and flow theory predictions. The modular architecture's engine independence and the behavioral adaptation transparency represent a novel architectural contribution relative to existing DDA implementations (see Table 1, Section 5.5) and provide a solid reusable foundation for future DDA research and development. This modular architecture is transferable in principle, but it has not been validated across multiple encounters. The experimental validation is narrower, limited to a single boss encounter that takes place in a single game, with  $n = 30$  participants, and should be interpreted accordingly.

**Author Contributions:** Conceptualization, K.J.S.; methodology, K.J.S., R.B. and S.N.; software, K.J.S.; validation, K.J.S. and R.B.; formal analysis, K.J.S. and R.B.; investigation, K.J.S.; resources, K.J.S.; data curation, K.J.S.; writing—original draft preparation, K.J.S.; writing—review and editing, R.B. and S.N.; visualization, K.J.S. and R.B.; supervision, R.B. and S.N. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding. The APC was funded by the authors.

**Institutional Review Board Statement:** This work is about the design and technical evaluation of a Dynamic Difficulty Adjustment (DDA) system. It is not a clinical study and does not report results from a clinical trial. The activity was conducted with adult volunteers, who gave their informed consent, using anonymous IDs, and with no personally identifiable information being collected. Under our institution's policies for minimal-risk educational research with anonymous data, ethical review was not required.

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in the study. Electronic informed consent was collected individually and stored alongside the collected data; documentation is available from the corresponding author upon reasonable request. The pub-

lic repository contains only data collected anonymously. Written informed consent has been obtained from all participants for the publication of anonymous results and illustrative figures derived from their data.

**Data Availability Statement:** The Papyrus script (DDA\_BossController.psc), Python live analyzer (dda\_live\_analyzer.py), survey results (anonymized), telemetry dataset, mod and save files are available at [https://github.com/kevinjacob10/DDA\\_Framework\\_Data](https://github.com/kevinjacob10/DDA_Framework_Data) (accessed on 15 July 2026). The Skyrim mod file requires a licensed copy of The Elder Scrolls V: Skyrim Special Edition through Steam and the SKSE run-time.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Zohaib, M. Dynamic difficulty adjustment (DDA) in computer games: A review. *Adv. Hum. Comput. Interact.* **2018**, *2018*, 5681652.
2. Csikszentmihalyi, M. *Flow: The Psychology of Optimal Experience*; Harper Perennial: New York, NY, USA, 1990.
3. Csikszentmihalyi, M. *Optimal Experience: Psychological Studies of Flow in Consciousness*; Cambridge University Press: Cambridge, UK, 1988.
4. Sherry, J.L. Flow and media enjoyment. *Commun. Theory* **2004**, *14*, 328–347.
5. Mortazavi, F.; Moradi, H.; Vahabie, A.-H. Dynamic difficulty adjustment approaches in video games: A systematic literature review. *Multimed. Tools Appl.* **2024**, *83*, 83227–83274. <https://doi.org/10.1007/s11042-024-18768-x>.
6. Romero-Mendez, E.A.; Santana-Mancilla, P.C.; Garcia-Ruiz, M.; Montesinos-López, O.A.; Anido-Rifón, L.E. The use of deep learning to improve player engagement in a video game through a dynamic difficulty adjustment based on skills classification. *Appl. Sci.* **2023**, *13*, 8249.
7. Guo, Z.; Thawonmas, R.; Ren, X. Rethinking dynamic difficulty adjustment for video game design. *Entertain. Comput.* **2024**, *50*, 100663.
8. Or, D.B.; Kolomenkin, M.; Shabat, G. DL-DDA: Deep-learning-based dynamic difficulty adjustment with UX and gameplay constraints. In *Proceedings of the IEEE Conference on Games (CoG), Copenhagen, Denmark, 21–24 August 2021*; IEEE: New York, NY, USA, 2021; pp. 1–8.
9. Larche, C.J.; Dixon, M.J. The relationship between the skill-challenge balance, game expertise, flow and the urge to keep playing complex mobile games. *J. Behav. Addict.* **2020**, *9*, 606–618.
10. Sweetser, P.; Wyeth, P. GameFlow: A model for evaluating player enjoyment in games. *Comput. Entertain.* **2005**, *3*, 3.
11. Sherry, J.L.; Lucas, K.; Greenberg, B.S.; Lachlan, K. Video game uses and gratifications as predictors of use and game preference. In *Playing Video Games: Motives, Responses, and Consequences*; Vorderer, P., Bryant, J., Eds.; Lawrence Erlbaum Associates: Mahwah, NJ, USA, 2006; pp. 213–224.
12. Corcos, A. Being enjoyably challenged is the key to an enjoyable gaming experience. *Socioaffect. Neurosci. Psychol.* **2018**, *8*, 1474668.
13. Hunicke, R. The case for dynamic difficulty adjustment in games. In *Proceedings of the ACM SIGCHI Conference on Advances in Computer Entertainment Technology, Valencia, Spain, 15–17 November 2005*; Association for Computing Machinery: New York, NY, USA, 2005; pp. 429–433.
14. Spronck, P.; Sprinkhuizen-Kuyper, I.; Postma, E. Difficulty scaling of game AI. In *Proceedings of the 5th International Conference on Intelligent Games and Simulation (GAME-ON), London, UK, 4–6 November 2004*; EUROSIS: Ostend, Belgium, 2004; pp. 33–37.
15. Spronck, P.; Ponsen, M.; Sprinkhuizen-Kuyper, I.; Postma, E. Adaptive game AI with dynamic scripting. *Mach. Learn.* **2006**, *63*, 217–248.
16. Capcom. *Resident Evil 4*; Video Game; Capcom: Osaka, Japan, 2005.
17. Hunicke, R.; Chapman, V. AI for dynamic difficulty adjustment in games. In *Proceedings of the AAAI Workshop on Challenges in Game AI, San Jose, CA, USA, 25–26 July 2004*; Association for the Advancement of Artificial Intelligence: San Jose, CA, USA, 2004; pp. 91–96.
18. Valve Corporation. *Left 4 Dead*; Video Game; Valve: Bellevue, WA, USA, 2008.
19. Andrade, G.; Ramalho, G.; Gomes, A.S.; Corruble, V. Dynamic game balancing: An evaluation of user satisfaction. In *Proceedings of the 2nd AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, Marina del Rey, CA, USA, 20–23 June 2006*; Association for the Advancement of Artificial Intelligence: San Jose, CA, USA, 2006; pp. 3–8.
20. Yannakakis, G.N.; Togelius, J. Experience-driven procedural content generation. *IEEE Trans. Affect. Comput.* **2011**, *2*, 147–161.

21. Pedersen, C.; Togelius, J.; Yannakakis, G.N. Modeling player experience in Super Mario Bros. In *Proceedings of the IEEE Symposium on Computational Intelligence and Games, Milan, Italy, 7–10 September 2009*; IEEE: New York, NY, USA, 2009; pp. 132–139.
22. Hagelbäck, J.; Johansson, S.J. Measuring player experience on runtime dynamic difficulty scaling in an RTS game. In *Proceedings of the IEEE Symposium on Computational Intelligence and Games, Milan, Italy, 7–10 September 2009*; IEEE: New York, NY, USA, 2009; pp. 46–52.
23. Silva, M.P.; Silva, V.N.; Chaimowicz, L. Dynamic difficulty adjustment on MOBA games. *Entertain. Comput.* **2017**, *18*, 103–123.
24. Rosa, M.P.C.; Castanho, C.D.; Silva, T.B.P.; Sarmet, M.M.; Jacobi, R.P. Dynamic difficulty adjustment by performance and player profile in platform game. In *Proceedings of Entertainment Computing—ICEC 2023, Bologna, Italy, 15–17 November 2023*; Springer: Singapore, 2023; pp. 3–16.
25. Zook, A.; Riedl, M.O. A temporal data-driven player model for dynamic difficulty adjustment. In *Proceedings of the 8th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, Stanford, CA, USA, 8–12 October 2012*.
26. Paraschos, P.D.; Koulouriotis, D.E. Fuzzy-logic-based dynamic difficulty adjustment for adaptive game environments. *Electronics* **2025**, *14*, 146.
27. Damastuti, F.A.; Firmansyah, K.; Arif, Y.M.; Dutono, T.; Barakbah, A.; Hariadi, M. Dynamic level of difficulties using Q-learning and fuzzy logic. *IEEE Access* **2024**, *12*, 137775–137789. <https://doi.org/10.1109/ACCESS.2024.3457801>.
28. Fuchs, R.; Gieseke, R.; Dockhorn, A. Personalized dynamic difficulty adjustment—Imitation learning meets reinforcement learning. *arXiv* **2024**, arXiv:2408.06818.
29. Fisher, N.; Kulshreshtha, A.K. Exploring dynamic difficulty adjustment methods for video games. *Virtual Worlds* **2024**, *3*, 230–255.
30. Bauckhage, C.; Drachen, A.; Sifa, R. Clustering game behavior data. *IEEE Trans. Comput. Intell. AI Games* **2015**, *7*, 266–278.
31. Drachen, A.; Canossa, A.; Yannakakis, G.N. Player modeling using self-organization in Tomb Raider: Underworld. In *Proceedings of the IEEE Symposium on Computational Intelligence and Games, Milan, Italy, 7–10 September 2009*; IEEE: New York, NY, USA, 2009; pp. 1–8.
32. Sifa, R.; Bauckhage, C.; Drachen, A. Profiling in games: Understanding behavior from telemetry. In *Social Interactions in Virtual Worlds*; Cambridge University Press: Cambridge, UK, 2018; pp. 337–374.
33. Swanson, L.; Gagnon, D.; Scianna, J.; McCloskey, J.; Spevacek, N.; Slater, S.; Harpstead, E. Leveraging cluster analysis to understand educational game player experiences and support design. *arXiv* **2022**, arXiv:2210.09911.
34. Harpstead, E.; Zimmermann, T.; Nagapan, N.; Guajardo, J.J.; Cooper, R.; Solberg, T.; Greenawalt, D. What drives people: Creating engagement profiles of players from game log data. In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play, London, UK, 5–7 October 2015*; Association for Computing Machinery: New York, NY, USA, 2015; pp. 369–379.
35. Bethesda Game Studios. Creation Kit Documentation. Available online: [https://en.uesp.net/wiki/Skyrim\\_Mod:Creation\\_Kit](https://en.uesp.net/wiki/Skyrim_Mod:Creation_Kit) (accessed on 21 April 2026).
36. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
37. PapyrusUtil—Modders Scripting Utility Functions. Available online: <https://www.nexusmods.com/skyrimspecialiedition/mods/13048> (accessed on 21 April 2026).
38. Bethesda Game Studios. *The Elder Scrolls V: Skyrim Special Edition*; Video Game; Bethesda Softworks: Rockville, MD, USA, 2016.
39. SKSE—Skyrim Script Extender. Available online: <https://skse.silverlock.org/> (accessed on 28 May 2025).
40. Rousseeuw, P.J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **1987**, *20*, 53–65.
41. Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.