

Implementation of Machine Learning-Based Enemy AI Adaptive to Player Skill in 3D Action Game

Hiroto Sawai

s1300055

Supervised by Maxim Mozgovoy

Abstract

In this study, we propose a dynamic difficulty adjustment (DDA) method that adapts to individual skills without compromising the player's immersion in a 3D action game. Conventional methods that rely on numerical parameter adjustments (increase or decrease in attack power and HP) have had the problem of unnatural adjustments and cannot accommodate individual player quirks. In this method, reinforcement learning and imitation learning are combined to construct a "Personalized Agent" that performs counter-behavior optimized for individual play styles by performing reinforcement learning on agents that have learned and imitated the behavioral characteristics of specific players. As a result of the verification experiment, the average win rate of players before and after learning improved from 24.4% to 53.3%, and they succeeded in creating a competitive state. In addition, 88.9% of the players said that they enjoyed playing against agents more after learning, indicating that this method is effective in both objective and subjective aspects.

1 Introduction

In recent years, action games characterized by high difficulty levels have been favored by many players. The sense of accomplishment that comes from overcoming a difficult challenge is one of the main attractions of the gaming experience. However, the perception of difficulty can vary greatly depending on the player's proficiency level and playstyle, resulting in a different experience. If the difficulty is too high, the player will feel strong stress and get frustrated, and if it is too low, it will lead to boredom [1]. This state, where 'challenge' and 'player's skill' are balanced and the player is immersed, is called the 'flow state' [2] and is an important indicator for achieving an excellent gaming experience.

Dynamic difficulty adjustment (DDA) has been widely studied as a technique to keep players in a flow state [3]. Most DDAs implemented in traditional commercial games are rule-based methods that dynamically increase or decrease numerical parameters such as enemy attack power, HP, and movement speed based on indicators such as the player's win rate and remaining HP. While this has the advantage of making it easier to adjust game balance

and game design, these methods have the following challenges:

Unnatural Adjustments: Blatant fluctuations in parameters can give players a sense of being "underestimated" and reduce immersion.

Non-response to individual quirks: Uniform adjustments such as attack power can make it difficult to adapt to a player's individual playstyle, such as "attack timing quirks" and "specific behavior patterns (e.g., heavy evasion, preference for ranged attacks)".

According to Sepulveda et al. [6], if the player feels that the difficulty of the game is intentionally adjusted in this way, it will reduce immersion in the game and discourage the player, so an effective DDA requires an approach in which the adjustment is not noticed by the player.

Therefore, in this study, we propose a DDA method that adapts to the player's individual play style using machine learning, which does not depend on parameter manipulation. In this method, by combining reinforcement learning and imitation learning, it learns the characteristics of a specific player's movements and constructs an agent that takes countermeasures according to their personality. The hypothesis of this study is that by adjusting the difficulty level by "behavior optimization" rather than "numerical adjustment", it is possible to create an antagonistic state optimized for individual skills without causing the player to detect the adjustment.

2 Related Work

Personalized Dynamic Difficulty Adjustment

Personalized Dynamic Difficulty Adjustment (PDDA) [4] is a method that uses machine learning to analyze player movements and dynamically adjust enemy strength based on player skill. PDDA builds agents that adapt to the player's individual abilities through the following process:

Imitation Learning: Construct an "Imitation-Agent" that records real player movements and uses imitation learning to mimic player movements.

Reinforcement learning with an imitation learning agent as an opponent: Reinforcement learning is performed against the built Imitation-

Agent to generate an agent that provides the optimal difficulty level for a specific player's quirks and level.

In [4], PDDA was implemented for a 2D fighting game, while in our study we will verify whether this method is effective in 3D action games based on this PDDA method.

3 Method

3.1 Environment

To achieve our goals, we have created a 1-vs-1 competitive 3D action game using Unity to verify the proposed method. As a learning framework, we used Unity ML-Agents Release 22 [5] and PyTorch [7].

The game developed in this experiment employs a third-person perspective, where players and agents engage in combat on a square, flat battlefield. Health points (HP) are displayed above each character's head as a bar, allowing players to visually monitor the situation. Both sides start with 100 HP, and attacks reduce the opponent's HP. The victory condition is "reducing the opponent's HP to 0."

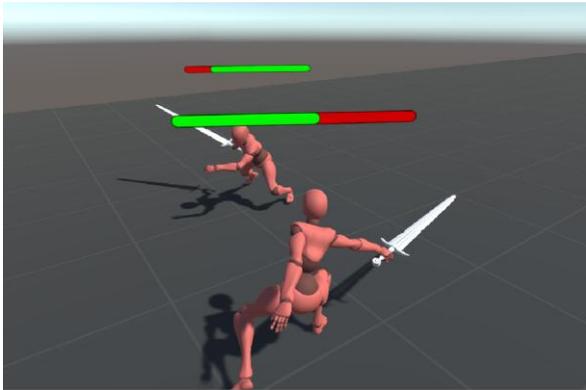


Figure 1. The game environment developed in Unity

Regarding character traits, the actions available to both players and agents are identical, as are the basic parameters such as movement speed, attack power, and hit rating. In terms of actions, four specific types are defined:

- Movement (forward, backward, left, right)
- Light / Heavy Attack (2 types of attacks)
- Avoidance Behavior (rolling).

3.2 Agent Design

Here, we define the types of agents and what they perceive and how they behave.

The learning flow of this study is based on the previous PDDA method, and the following three types

of agents are used.

- **Base Agent:** The first enemy agent players face, demonstrating average-level abilities while teaching the game's core mechanics.
- **Imitation Agent:** An agent that reproduces and learns the recorded player's behavior.
- **Personalized Agent:** An agent that plays against Imitation Agent and learns through reinforcement learning.

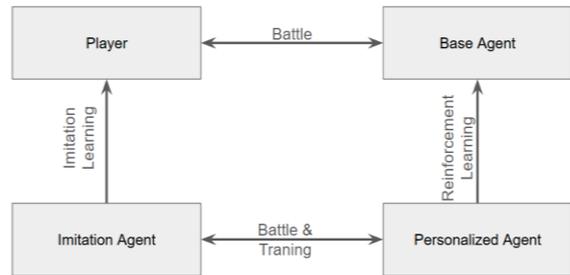


Figure 2. Agents Architecture

The models of these three types of agents are the same, and the Imitation Agent and Personalized Agent are trained based on the Base Agent. The agent observes the target's HP, distance from the target, and its own movement speed. From the start of the game, the agent always faced the direction of the target.

The learning flow of this study is based on the method of PDDA, which is a previous study, and consists of the following three stages.

- **Phase 1: Base Training**
Reinforcement Learning (PPO) is performed on Base Agents to learn the basic elements of a match, such as approaching players, hitting attacks, and defeating players. The number of learning steps was 3.0×10^7 steps.
- **Phase 2: Imitation Learning**
Play 5 matches against real players with Base Agent, record the player's "state" and "behavior" information at that time, then use Behavior Cloning (BC) and GAIL to imitation learning and create Imitation Agent.
- **Phase 3: Personalization**
Perform reinforcement learning of the Personalized Agent with the Imitation Agent as an opponent. At this time, if the reinforcement learning agent's win rate is less than 50%, a positive reward will be given, and if it is more than 50%, a negative reward will be given when the opponent is defeated when hitting an attack (Table 1). This enables the exploitation of specific

players' weaknesses or the optimization of agents for the same level. Both imitative learning and reinforcement learning performed 5.0×10^4 steps.

Table 1. Agent reward

Distance from the target within 5 meters	+0.01
Time lapse	-0.01
Hit the attack	+1.0 / -1.0
Defeat	+5.0 / -5.0

In this reinforcement learning approach (PPO), the neural network does not directly manipulate numerical parameters such as the agent's movement speed or attack cooldown time. Instead, it takes the current observation as input and outputs a policy—the probabilities of selecting defined actions (move, attack, avoid) in response. Rewards are used to update the weights of this network. In other words, if the action of “attack” yields a favorable outcome under certain conditions, the probability of selecting “attack” again in that situation increases; conversely, if it yields an unfavorable outcome, the probability decreases.

3.3 Experiment

This experiment involved 9 university and graduate students aged 19 to 23 (8 males, 1 female) who regularly play video games. They first played 5 matches with the Base Agent, and then 5 matches with the Personalized Agent after training. The number of wins and losses was recorded as an objective evaluation, and a questionnaire survey was conducted as a subjective evaluation.

4 Result

4.1 Match Result

In order to verify the effectiveness of the agent constructed by the proposed method, a head-to-head test was conducted between the agent and the player before training (Base Agent) and after training (Personalized Agent). Figure 3 shows the player's win rate against each agent.

As a result, the win rate against the agent after learning changed from 24.4% to 53.3%. Regarding individual win rates, players who won before learning had a lower win rate for post-learning agents, and players who were losing increased for post-learning agents.

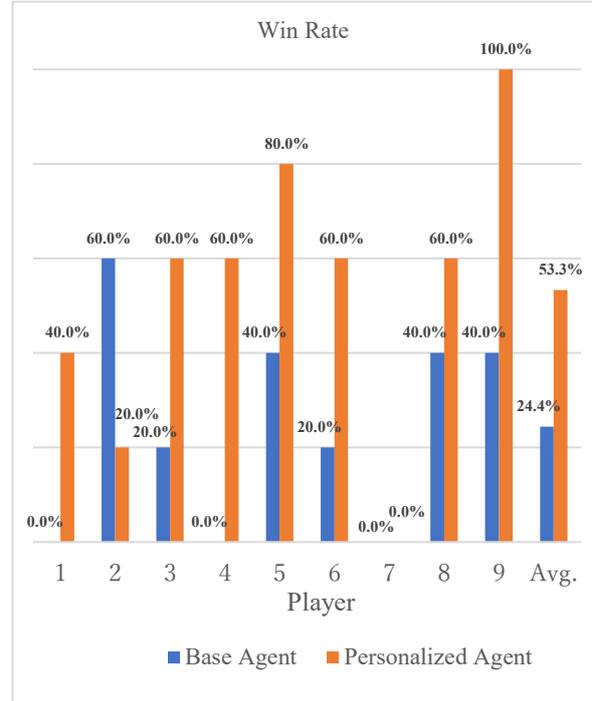


Figure 3. Results of the match against the agent

4.2 Agent Behavior

Here, we describe the changes in behavior patterns of enemy agents before and after training.

The Base Agent approached the player immediately after the game started and launched an attack. At this time, after the attack, the agent took an evasive action, and as soon as the cooldown expired, he would attack again.

On the other hand, Personalized Agents approach the player immediately after the game starts and launch an attack, but many of them show gaps from attack to evasive action. In addition, for players who used to use rolling a lot in the fight against the Base Agent, there was an increase in the number of attacks immediately after the player's evasive action.

4.3 Survey Result

After the match, 9 subjects were surveyed about "how strongly they felt the agent" and "which agent they enjoyed playing against" before and after the study, and the former was rated on a 5-point scale, and the latter was conducted in a two-choice format. The aggregate results are shown in Tables 2 and 3.

Table 2. Survey Results on the Strength of Enemy Agents

	Base Agent	Personalized Agent
Too Hard	11.1%	0%
Hard	55.6%	22.2%
Moderate	33.3%	44.4%
Easy	0%	33.3%
Too Easy	0%	0%

Table 3. Survey on which agent was more enjoyable

Base Agent	11.1%
Personalized Agent	88.9%

Regarding perceived strength, in their first match against the AI, 66.7% of players answered "difficult (55.6%)" or "very difficult (11.1%)", while only 33.3% of players answered "just right," indicating that the initial agent was at a high difficulty level for many players. In the match against the AI after reinforcement learning, the number of players who answered "just right" increased to 44.4%, the number of players who answered "difficult" decreased to 22.2%, and a new layer of "easy (33.3%)" answers appeared.

Furthermore, in terms of player enjoyment, 88.9% of players said they enjoyed playing against the agents after the learning process.

5 Discussion

In this study, we will evaluate whether DDA is appropriate from both objective and subjective perspectives by questionnaires about whether the difficulty level is appropriately adjusted according to the player, and by taking questionnaires on the player's winning or losing game and experiencing the game.

Of the nine participants in the experiment, 33.3% had a good initial difficulty, compared to 44.4% of the post-learning agents.

Also, most of the people who answered that it was difficult at the beginning answered that it was just right for an agent after learning.

In terms of win rate, the player's win rate was 24.4% for the first agent, compared to 53.3% for the post-learning agent. This is thought to be due to the fact that agents have increased their behavior of showing gaps after learning, and the player's attacks have increased. On the other hand, there were also behaviors that attacked immediately after the player dodged, so it can be assumed that the agent was adapting to the player's behavior without becoming too weak.

Based on this suggestion, it can be evaluated that a certain level of DDA can be performed appropriately.

However, since it took about 5 minutes to imitate the player's movements and complete reinforcement learning, how to reduce this time to run in real time is a challenge in the future.

6 Conclusion

In this study, we proposed a dynamic difficulty adjustment (DDA) method for 3D action games using machine learning that does not rely on rule-based or numerical parameter manipulation in 3D action games. By combining reinforcement learning and imitation learning, we built a personalized agent that learns the behavioral characteristics of a specific player and takes countermeasures accordingly.

As a result of the experiment, the average win rate of players against agents using the proposed method changed from 24.4% before learning to 53.3% after learning and succeeded in creating a competitive situation with a win rate of around 50%. The survey also confirmed that 88.9% of players "enjoyed" playing against agents after learning, confirming an improved subjective gaming experience. These results support the study's hypothesis that difficulty adjustment through "behavioral optimization" rather than "numerical adjustment" can provide a gaming experience that adapts to individual skills without compromising the player's immersion.

Future challenges include shortening the learning time for individual adaptation. Currently, it takes about 5 minutes to complete reinforcement learning by mimicking the player's movements, and further efficiency is a challenge in the future to complete adaptations in real time during gameplay.

References

- [1] Johan Hagelback and Stefan J Johansson, "Measuring player experience on runtime dynamic difficulty scaling in an RTS game.", IEEE Symposium on Computational Intelligence and Games, pages 46–52, 2009.
- [2] Mihaly Csikszentmihalyi, "Flow: The psychology of optimal experience", volume 1990. Harper & Row New York, 1990.
- [3] Mohammad Zohaib, "Dynamic difficulty adjustment (DDA) in computer games: A review." Advances in Human-Computer Interaction, 2018.
- [4] Ronja Fuchs et al. "Personalized Dynamic Difficulty Adjustment – Imitation Learning Meets Reinforcement Learning", IEEE Conference on Games, 2024.
- [5] Unity ML-Agent <https://github.com/Unity-Technologies/ml-agents>
- [6] Gabriel K. Sepulveda et al. "Exploring

- Dynamic Difficulty Adjustment in Videogames”, IEEE CHILECON, 2019.
- [7] Adam Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”, Advances in Neural Information Processing Systems, 2019.