Genetic Encoding of Synergies: Coevolving Traits, Behaviors, and Strategies for Game AI

Lucas Schurer[1], Luis A. L. Silva[1], Cesar T. Pozzer[1], Joaquim V. C. Assunção[1]

[1] Universidade Federal de Santa Maria (UFSM) – Santa Maria, RS – Brasil

{lschurer,luisalvaro,pozzer,joaquim}@inf.ufsm.br

Abstract. Introduction: *Several games use data to train machine learning models, which requires extensive gameplay before creating the model. Moreover, these models are designed to be strong in the game, rather than adapting to the player's level.* ***Objective:*** *To address this problem, this work focuses on coevolution through the embedded use of Genetic Algorithms (GAs) to adapt the behaviors of game characters to those of human players.* ***Methodology:*** *We describe how the various elements (characters' traits, weapons, and skills) of a developed 3D game were projected and implemented, showing a concrete instance of how to develop similar games. Then, this work details the mapping of these elements to the GAs so that groups of characters represent populations, and generations represent the rounds of a match. To evaluate the coevolution of agents with the proposed adaptation method, the combined use (or synergies) between weapons, traits, behaviors, and strategies are analyzed in diverse scenarios.* ***Results:*** *The experiments show that characters generated with GAs adapt effectively to different types of opponents and exhibit coherent strategic behavior, confirming the potential of the coevolution method to produce dynamic and responsive game agents.*
Keywords *Coevolution, Genetic algorithms, Game Adaptation, Games*

1. Introduction

Replayability refers to the possibility of playing a game multiple times. In [Krall and Menzies 2012], playability is described as the ability of an activity to be fun. Replayability is also described as how much a person can enjoy an activity until it stops being fun. Games like *Minecraft*, *Divinity: Original Sin*, *League of Legends*, and *Final Fantasy XIV Online* can provide several hours of content. This is due to their ability to create different worlds, the various ways to complete the game, as well as their competitive and social aspects. To increase the *replay value*, or replayability, several games use strategies based on Artificial Intelligence (AI). This type of generation encompasses those aspects of a game that impact gameplay.

AI techniques can be used to create more engaging game characters while avoiding rigid and predictable behavior. Techniques can be used to create characters that play optimally, seeking the best possible solutions for a given game scenario. For instance, the review at [Justesen et al. 2019] details solutions that use Deep Learning to generate NPCs capable of playing a game well, although they do not frequently aim to play it enjoyably. The authors in [Galway et al. 2008] review AI techniques that utilize neural networks, evolutionary algorithms, and other hybrid AI approaches. While these methods may not always lead to the best decisions in a game scenario, they often present believable and interesting behaviors for the player.

Genetic algorithms (GAs), inspired by natural evolution, are a potent tool for tackling optimization problems across diverse domains [Vie et al. 2021]. In game development, GAs offer the potential to automatically generate diverse and adaptive game characters, thereby enriching player experiences and expanding the possibilities for dynamic gameplay. Despite recent works using GAs for games, they are mostly focused on designing the environment, level challenges, and even the game story using Procedural Content Generation [Cook et al. 2016]. In contrast, this paper examines the generation of adaptive game characters using a *coevolutionary* approach with GA. By coevolving multiple features, such as character characteristics based on their opponents, we demonstrate that GAs produce compelling non-player characters that react dynamically to player actions. Our work focuses on coevolving the characters themselves in a group environment, adapting traits, behaviors, and interactions according to the opponent's team behaviors.

This work demonstrates how to coevolve characters to exhibit adaptive game behaviors, allowing them to respond to various game situations. To support the analysis of character adaptation capabilities, we explore the different characteristics of a developed 3D game in which character groups battle in multiple rounds. The paper details how the various game elements were developed, contributing to the design and implementation of computer games. Importantly, we show how the game characteristics were mapped to the GAs so that populations are represented by groups of characters, and generations represent the rounds of a match. In the experiments (See Section 5), character groups generated with GAs are generated according to three strategies: *offensive*, *defensive*, and *balanced*. In these tests, character groups generated with GA battle in multiple matches against a *random group*, where all characters in the group have all their elements completely randomized each new round, and a *static group*, where no character changes its behavior throughout the match rounds, maintaining the same initialized characteristics. Moreover, matches between two *GA-generated character groups* are executed, where the adaptability of the characters is observed in an environment where both competitors react to their opponents. In doing so, the fitness of these characters is built through a process of competitive coevolution [Elfeky et al. 2021] [Rawal et al. 2010], taking into account data from ally and enemy groups.

2. Related Works

Most related works focus on the procedural generation of game characters and adjusting game difficulty based on players' profiles. These works assess aspects of game evolution through GA techniques, from simpler game genres to more complex RTS games. Similar to [Dockhorn and Kruse 2017], our work investigates competitive coevolution between groups of ally and enemy characters. Unlike us, however, they analyze the evolution of behavior trees linked to the characters' decision-making. In [Liu et al. 2016], GAs are used to optimize the decision-making process of complex RTS games. Again, evolution does not seek to modify the characters' decision-making in our work since we aim to evolve other character characteristics. In [Zaidan and Góes 2016], character groups are used as input to the GAs while searching for a balance between game elements. Our work is not focused on balancing the game, but rather on adapting the character groups. Similar to [Pereira et al. 2021], character characteristics are explored in the evolutionary process. However, their work does not investigate the procedural generation of the character groups as we do through a coevolution process. In [Segundo et al. 2016], [Garcia et al. 2018], and [Weber and Notargiacomo 2020], the game difficulty is adjusted dynamically, adapting game elements to ensure a better player experience. Despite this, these studies do not investigate multiple character groups generated with GA that compete with each other. As in [Faria et al. 2019], our fitness function is also composed of several weight components. In their work, however, a player profile determines these weights, while our set of weights is defined by the strategy used by the character groups.

3. A Developed Game

To support the game's adaptation proposal detailed in this work, we detail the various characteristics of a 3D game implemented in Unity. In this game, the rounds of dispute between two opposing groups of characters are related to the generations created throughout an evolutionary process. This adjustment causes the groups of opponents to adapt, forcing the opposing group to use different strategies.

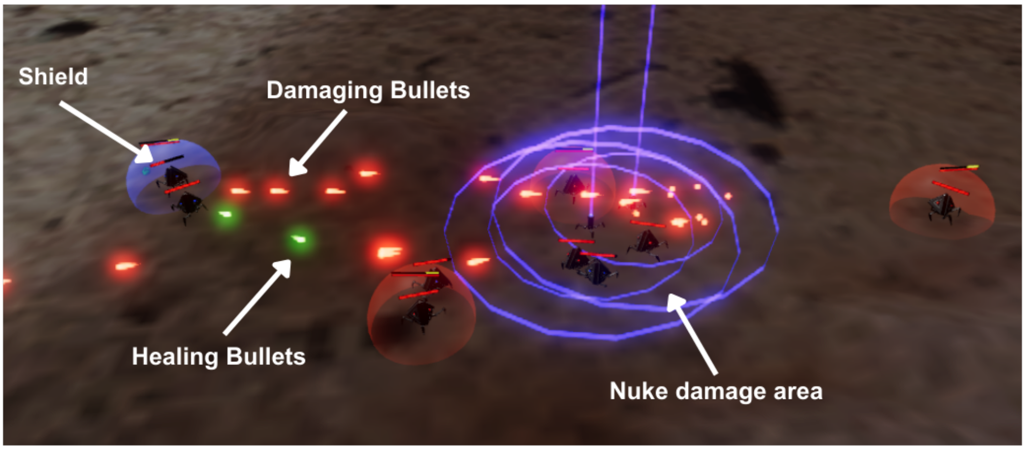

Figure 1. Bird eye view of the 3D shooting game implemented in this work.

3.1. Game Characters: Allies and Enemies

All game characters have *weapons*, responsible for interactions with the character's life points; *behaviors*, defining the character's decisions throughout a round; and *traits*, representing particular characters' characteristics. Defining the way characters interact with the game environment, the characters were modeled by the following attributes: *Health*, the damage a character can take before dying; *MovementSpeed*, how fast a character moves around the map; *HealingMultiplier*, the multiplier used when healing an ally, increasing or decreasing the amount healed; *CommonDamageMultiplier* and *ExplosiveDamageMultipliers*, the multipliers used when dealing with common and explosive damage, respectively, increasing or decreasing the damage depending on such multipliers; *CommonDamageReductionMultiplier* and *ExplosiveDamageReductionMultiplier*, the multipliers used when taking common and explosive damage, respectively, decreasing or increasing the damage taken according to the multipliers.

Events linked to game characters are observed during the rounds of a match. This results in a set of statistical measures regarding the characteristics of a given character.

Called *stats*, these measures guide the evolutionary process. At the end of a round, we observe how the character interacted with the environment and with other characters through the following stats: *DamageDealt*, the damage dealt by the character to enemies during the round; *HealingDealt*, the healing dealt by the character to allies during the round; *DamageBlocked*, the damage blocked by the Shield weapon for the round; *Alive*, represents the life status of a character at the end of the round.

3.2. Weapons: Damaging Bullets, Nuke, Healing Bullets, and Shield

The *Damaging Bullets* create projectiles directed at an enemy, dealing a common type of damage on impact. The low cooldown, in addition to the moderate damage, results in an extremely effective weapon against characters with healing bullets. This setup deals with more damage than the Healing Bullets weapon can recover. However, it is ineffective against the shield due to its high durability. Shield blocks the projectiles created by the Damaging Bullets.

The *Nuke* creates an immobile area at an enemy position. After an activation period, it causes explosive damage in that area, hitting all enemies present. A Nuke has a high cooldown and damage, which allows it to reduce the enemies' health. The Nuke's cooldown allows enemies to use Healing Bullets to recover their damage. On the other hand, the damage imposed by the Nuke significantly reduces the Shield's durability.

The *Healing Bullets* create projectiles directed at an allied character, restoring the character's health on impact. Projectiles are not intercepted by enemies. This weapon has a low cooldown, making it effective against Nukes. It can recover the character damage while the enemy weapon is on cooldown. The health recovered, however, cannot compete with Damaging bullets.

The *Shield* creates a protective and movable sphere around the character. The Shield's durability is reduced when damage is taken, and it is destroyed when its durability is zero. This weapon blocks projectiles created by Damaging Bullets. The Shield durability is significantly reduced when hit by the Nuke's explosive damage.

3.3. Character Behaviors

Character behaviors model the characters' decision-making, taking into account their health and their relationships with others. Non-deterministic Finite State Machines (FSMs) were implemented to model these behaviors, allowing the introduction of variability in the actions executed by ally and enemy characters. The modeled character behaviors are the following:

- Aggressive: it seeks only to attack enemies, not considering the character's life or the target when making a decision;
- Defensive: it gives preference to staying in front of allies. This behavior encourages characters to protect nearby allies by intercepting attacks.
- Tactical: It is a complex behavior that considers the enemies' and characters' life points, where characters run away from enemies or search for shielded allies when their health is low. This behavior can allow the characters to flank enemies and give preference to attacking those who have shields;
- Support: It is also a complex behavior that takes into account the life points of allies and characters, where characters flee from enemies or seek out allies to

protect themselves. When used, characters can also assume a defensive posture against allies by intercepting projectiles.

3.4. Character traits

Traits are responsible for positively or negatively adapting character attributes. A character can use only two traits at any given time. This implementation decision encourages strategic trait choice. Moreover, it simplifies the assessment of how the chosen traits affect the match.

A total of 17 traits were implemented. Here are some of them with relevance for our evaluation experiments: *MoreHealth*, *MoreMovementSpeed*, *MoreHealingMultiplier*, *MovementSpeed*, *HealingMultiplier*, *MoreCommonDamageMultiplier*, *CommonDamageMultiplier*, *MoreExplosiveDamageMultiplier*, *ExplosiveDamageMultiplier*, *CommonDamageMultiplier*, and *CommonDamageReductionMultiplier*.

4. Coevolution of Character Groups through Genetic Algorithms

GAs are employed in the developed game to adapt characters' groups to their opponents. The aim is to create less predictable characters and, consequently, to develop a more dynamic game environment, improving the player experience. In this work, we analyzed the adaptation of both allies and enemies. This was modeled as a *coevolution* process [Elfeky et al. 2021] [Rawal et al. 2010].

Coevolution is the simultaneous evolution of interdependent populations, where an individual's fitness is influenced by direct competition with others. As strategies evolve to increase one population's fitness, opponents tend to develop counter-strategies in response.

The GAs are represented by *generations*, which comprise the game's *population*. Such *population* is defined by the set of characters belonging to the same team. The *generations* of these populations are represented by the rounds of a match. Allies and enemies are created in each round of dispute. Both groups have the same randomly generated characters in the first game round. Weapons, behaviors, and traits are randomized from all implemented options. Then, characters are determined by selecting the best individuals from the previous generation.

The character is responsible for interacting with the environment by having its characteristics called *chromosomes*. Chromosomes are made up of *genes*. Thus, the character's weapons, behaviors, and traits are mapped to such *genes* as detailed in Figure 2. The genes associated with a game character allow adding, removing, or changing the game components. Ultimately, the character's genes form a set of characteristics that determine its actions in a specific game environment.

After defining the environment, population, character, and genes, we establish a *fitness* metric for how well a given character performed during its lifetime. Fitness is also responsible for directing the *evolution* and *selection* of characters, which are chosen based on the fitness value. In this work, the character's fitness is made up of the following factors:

- *Stat*: A performance evaluation involving Damage Dealt, Healing Dealt, Damage Blocked, and Alive collected throughout the match;

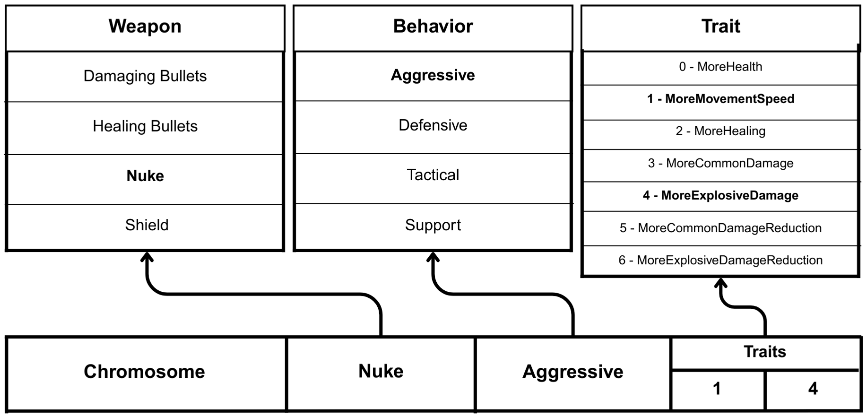

Figure 2. The character's chromosome.

- *Weight*: A relevance estimate associated with the stat and modeled between 0 and 1 in this work. Determines how important a given stat is to building a character's total fitness;
- *Maximum stat value across populations*: The maximum value found for a given stat across all pairs of populations.

A set of weights, representing a strategy, is individual to a given population, where fitness weights are unchanged throughout the match. For instance, a population can assign a higher fitness weight to the *DamageDealt* stat, resulting in a more offensive strategy where characters imposing more damage have higher fitness. Thus, different populations may employ different strategies, resulting in distinct game styles among populations. The sum of the weight values of a strategy must be equal to 1 so that the maximum character fitness is normalized to 1. In this work, the characters' fitness is defined as follows.

$$f = W * (S/S_{max})$$

Where W is the set of weights that determine the population's strategy, S the character's stats, and S_{max} the set of stats with the maximum value found between two populations.

Importantly, the *Maximum stat* value across populations is used for the *population coevolution* [Elfeky et al. 2021] [Rawal et al. 2010]. This allows characters to compete whenever their populations are varied in the game, avoiding comparisons only between characters from the same population. Thus, a high value in a certain character stat can result in low fitness values linked to characters from the opposing population.

The *selection* identifies parents who are fit to breed characters for the next generation. This selection is based on the characters' fitness values. Selection uses a roulette and tournament method randomly. After it, *crossover* is performed, combining the chromosomes of the selected parents to generate a new character. This work used a k-point crossover in the *crossover*. The *mutation* operator creates variety after crossover, seeking to prevent premature convergence, where the population becomes stuck in an

Table 1. Strategies and their initial weights.

Property/Strategy	Offensive	Defensive	Balanced
DamageDealt	0.7	0.0	0.44
HealingDealt	0.0	0.4	0.23
DamageBlocked	0.0	0.2	0.1
Alive	0.3	0.4	0.23

unacceptable solution, by randomizing the characters' genes. The mutation operator is applied only to the children generated by *crossover*, which have a chance of being mutated. All genes in a child with a mutation are randomly rearranged, resulting in a new combination of weapons, traits, and behaviors. *Elitism* is also employed, preserving the character with the best *fitness* from the current generation to the next.

At some point, the GA needs to stop creating new generations. This *stop condition* is the maximum number of generations. As our game rounds are mapped to these generations, the stop condition is the end of each disputed match.

5. Experiments and Results

To understand the characters' behaviors resulting from the GA executions, we conducted three major experiments. These experiments involved matches played to evaluate the adaptation capacity of characters generated using GAs. These matches are considered different competitors. First, the GA competitor played against a stochastic opponent. Second, the GA competitor played against a static opponent. Third, the GA competitor played against another instance of itself as the opponent. In each experiment, character groups played 33 matches, each consisting of 100 rounds. Each round contained 20 characters from each group and lasted 60 seconds.

In all scenarios, character groups generated with GA are tested using tournament and roulette selection methods, having one of three strategies: Offensive, Defensive, and Balanced. Each strategy has a particular focus, which is determined by defining weights for each stat that contributes to the character's fitness. The initial weights for each of the stats are presented in Table 1. The offensive and defensive strategies prioritize an extreme playing style, while the balanced one uses the set of weights described earlier in a balanced way. All character groups generated by GA use the same hyperparameters, such as mutation rate, elitism, the number of parents selected for crossover, and tournament size for groups with tournament selection. These values were adjusted empirically.

In the next sections, we detail the combats carried out and the combination of weapons, traits, and behaviors used by the groups generated by GA. Importantly, we use the term *synergy* for the combination of offensive or defensive weapons and traits for a given type of damage. That way, **offensive synergies** encompass the following pairs (weapon, trait): (Nuke, More Explosive Damage) and (Damaging Bullets, More Common Damage). Similarly, **defensive synergies** are: (Healing Bullets, More Healing), (Healing Bullets, Explosive Damage Reduction), (Healing Bullets, Common Damage Reduction), (Shield, Common Damage Reduction), and (Shield, Explosive Damage Reduction). The *Strategy* describes the collection of synergies according to their strategy. Hence, the **offensive strategy** is the character group that uses any offensive synergy, and the **defensive strategy** is the character group that uses any defensive Synergy.

5.1. GA vs. Random

Matches between character groups generated with GA against randomly generated groups are used to evaluate the characters' ability to adapt to a completely unpredictable enemy without a predefined strategy. Six fights covered the two selection methods and the three possible strategies. Due to the random nature of the use of weapons, behaviors, and traits in the randomly created character group, we show only information from character groups generated with GAs to make visualization simpler (Figure 3).

Using the offensive strategy, the character groups generated by GA achieved the expected distribution of weapons, with greater use of Damaging Bullets and Nuke weapons. The aggressive behavior was the most commonly used of the two selection methods. The character groups alternated between offensive weapons throughout the match. Depending on the weapon used, they also created offensive synergies, using the More Movement Speed trait in conjunction with More Common Damage or More Explosive Damage. For both selection methods, an average of 71% of match deaths were attributed to the Random character group.

Despite using more Damaging Bullets for all three strategies, we notice the lack of convergence for a specific weapon. That is expected since the opponent executes random synergies, and it is only natural for the best competitor to take different actions. Such behavior is well known from simple games such as rock-paper-scissors, in which the proved Nash-Equilibrium is a simple random, equally probable action [van den Nouweland 2007]. Figure 3 shows the synergies achieved in this experiment.

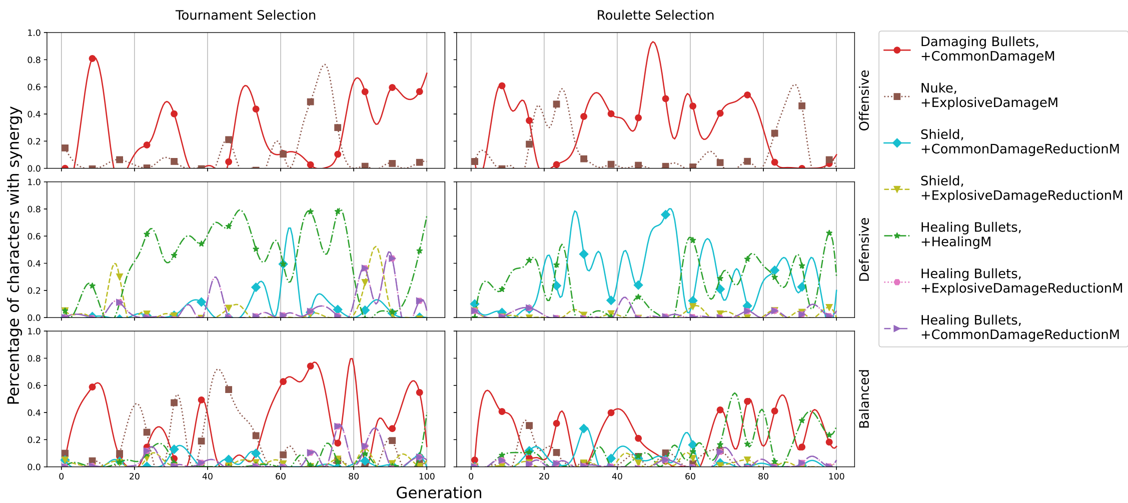

Figure 3. Overview of GA Synergies per Generation vs. Random.

5.2. GA vs. Static

Using selection methods and offensive, defensive, and balanced strategies, fights between character groups generated by GA are carried out against an opponent group initialized randomly. However, the opponent remains the same throughout the rounds. Moreover, all fights used the same seed in the initialization of the static character group. The goal is to observe how character groups generated with GA behave when faced with enemies that do not change their play style. This allows the GA character groups to converge towards a single play style. Thus, it is expected to converge to a sub-optimal solution, using the same weapons, behaviors, and traits to exploit the opponent's static behavior. The static character group covers the use of the Nuke, Healing Bullets, and Shield weapons, with a low use of Damaging bullets. It also predominantly uses the Support, Tactical, and Defensive behaviors, with the aggressive behavior explored weakly.

Unlike the GA vs. Random experiment, where there was no convergence, there was little preference for using the most common/default weapon. However, the GA characters successfully exploited the static opponents in this GA vs. Static experiment.

With the **offensive strategy**, character groups generated by GAs converged on using the Damaging Bullets weapon, and employed it throughout the match. The character group that used the tournament method predominantly exhibited aggressive behavior, while the tournament method used all behaviors in a balanced way. The character group with the roulette method used the More Healing trait and the More Common Damage trait throughout the match. On average, 90% match deaths were attributed to the Static character group.

With the **defensive strategy**, character groups generated by GA converged on the Healing Bullets weapon and Defensive behavior, presenting a defensive synergy with the Healing Bullets weapon and the More Healing trait. This scenario ended with 87% of match deaths attributed to the static group.

With the **balanced strategy**, a significant difference in weapon choice between the two selection methods was observed, with the tournament method using the Damaging Bullets weapon and the tournament method using the Nuke weapon for most of the match. However, both selection methods tended to use Aggressive Behavior, and both converged on a type of weapon. The tournament character group was responsible for 99.6% of the opposing group's deaths, with the roulette method responsible for 89.4% of match deaths. This difference can be attributed to the character group's use of the Explosive Damage Reduction trait with the tournament method. This is because the static character group used the Nuke weapon more often than the Damaging Bullets weapon, which increased the characters' survival in the group generated by GA.

5.3. GA vs. GA

To evaluate the adaptability of characters in dynamic scenarios, we conducted a set of experiments in which both competing character groups were generated through Genetic Algorithms (GAs). These matches aimed to assess how well the algorithm adapts when faced with an opponent that is also evolving in real-time. We explored all combinations of the three proposed strategies (Offensive, Defensive, Balanced) with both selection methods (Tournament and Roulette), resulting in 21 unique combat configurations.

Character groups demonstrated distinct patterns of adaptation. In combats, where both groups used the offensive strategy, there was frequent changing between Damaging Bullets and Nuke weapons, as well as between different synergies (e.g., Damaging Bullets with More Common Damage, Nuke with More Explosive Damage). The aggressive behavior was predominantly observed in these matches, with both sides adjusting their tactics throughout the match in response to the opponent's actions. Furthermore, synergy alternation was consistent, indicating that both GAs effectively identified and responded to the opponent's dominant traits.

When character groups with offensive strategies faced defensive ones, offensive groups generally preferred Damaging Bullets, while defensive groups made balanced use of Healing Bullets and Shield weapons. While synergies were present on both sides, defensive groups occasionally showed less synergy switching, likely due to converging quickly to effective countermeasures. Probably, a defensive group failed to adapt/improve after an initial configuration due to fast defeats, which prevents exploration of alternatives.

Balanced strategy groups showed behaviors more aligned with offensive strategies. These groups often employed the aggressive behavior and offensive synergies while keeping a moderate use of defensive elements. Against offensive or defensive opponents, balanced groups tended to adopt a hybrid weapon distribution and showed a higher strategy variability over time. Matches among balanced groups also demonstrated regular switching between weapons and traits, which reinforces the adaptability of the GA in less polarized configurations. Regarding the behavior usage, aggressive behavior was the most frequently adopted across all strategy match-ups. Defensive and support behaviors were less prevalent, even in defensive strategy groups. This might suggest a stronger evolutionary bias toward offensive play styles under the tested fitness configurations, which is expected since the agents need to attack in order to defeat opponents.

In summary, the GA vs. GA experiments confirm the effectiveness of the coevolutionary approach in dynamic, competitive environments. The observed weapon and synergy switching behavior highlights the algorithm's capability to adapt to continuously evolving opponents. Furthermore, the experiments emphasize the GA's tendency to converge on aggressive and high-damage strategies, especially when such strategies result in faster or more decisive victories.

6. Concluding Remarks

This work explores the generation of adaptive characters using genetic algorithms (GAs) in a group-vs-group battle environment. The coevolutionary approach investigated offers a relevant solution for game developers seeking to enhance the gaming experience and, consequently, increase replayability. A newly developed 3D game is presented throughout the work, serving as a testing environment for evaluating various adaptive character traits. The study also details how the game's elements are mapped to the GAs, which drive the coevolutionary process responsible for character adaptation.

The generated characters are evaluated in combats that include all combinations between randomized and static groups, as well as groups generated using GAs. Two selection methods — tournament and roulette —are tested, along with offensive, defensive, and balanced strategies. The test results revealed the characters' adaptability resulting from seamless GA executions.

Future work will analyze matches disputed between adaptive character bots and human players, aiming to further evaluate the impact of adaptation on the game's enjoyability and replayability. The approach to adaptive character generation discussed here can also be analyzed in different types of games. Lastly, other AI techniques can be used to generate adaptable characters, aiming to understand and combine other character-generation capacities with the techniques investigated in this work.

References

Cook, M., Eladhari, M., Nealen, A., Treanor, M., Boxerman, E., Jaffe, A., Sottosanti, P., e Swink, S. (2016). Pcg-based game design patterns.

Dockhorn, A. e Kruse, R. (2017). Combining cooperative and adversarial coevolution in the context of pac-man. pages 60–67, New York, NY.

Elfeky, E., Elsayed, S., Marsh, L., Essam, D., Cochrane, M., Sims, B., e Sarker, R. (2021). A systematic review of coevolution in real-time strategy games. *IEEE Access*, 9.

Faria, F. G., Pereira, L. T., e Toledo, C. F. M. (2019). Adaptive generation of characters for tabletop role playing games. pages 39–46.

Galway, L., , Charles, D. K., e Black, M. M. (2008). Machine learning in digital games: A survey. *Artificial Intelligence Review*, 29:123–161.

Garcia, B. E. R., Crocomo, M. K., e Andrade, K. O. (2018). Dynamic difficulty adjustment in a whac-a-mole like game. pages 88–888.

Justesen, N., Bontrager, P., Togelius, J., e Risi, S. (2019). Deep learning for video game playing.

Krall, J. e Menzies, T. (2012). Aspects of replayability and software engineering: Towards a methodology of developing games. *Journal of Software Engineering and Applications*, 5(7):459–466.

Liu, S., Louis, S., e Ballinger, C. (2016). Evolving effective micro behaviors in real-time strategy games. *IEEE Transactions on Computational Intelligence and AI in Games*, 8.

Pereira, L. T., Viana, B. M. F., e Toledo, C. F. M. (2021). Procedural enemy generation through parallel evolutionary algorithm. pages 126–135.

Rawal, A., Rajagopalan, P., e Miikkulainen, R. (2010). Constructing competitive and cooperative agent behavior using coevolution. pages 107–114.

Segundo, C. V. N., Calixto, K. E. A., e De Gusmão, R. P. (2016). Dynamic difficulty adjustment through parameter manipulation for space shooter game.

van den Nouweland, A. (2007). Rock-paper-scissors a new and elegant proof. *Economics Bulletin*, 3(43):1–6.

Vie, A., Kleinnijenhuis, A. M., e Farmer, D. J. (2021). Qualities, challenges and future of genetic algorithms: a literature review.

Weber, M. e Notargiacomo, P. (2020). Dynamic difficulty adjustment in digital games using genetic algorithms. pages 62–70.

Zaidan, T. e Góes, L. F. W. (2016). Evolvestone an evolutionary generator of balanced digital collectible card games. In *Brazilian Symposium on Computer Games and Digital Entertainment*.