Dynamic Difficulty Balancing of Strategy Games through Player Adaptation using Top Culling

Alex Fernandes da V. Machado[2], Cristiano Gomes Duarte[1], Heudes Eduardo Rogério[1], Lindomar Matos de Paulo[1], Wenderson Cirilo Silva de Paula[1], Esteban Walter Gonzalez Clua[2], Timothy James Rogers[3]

[1]Instituto Federal de Educação, Ciência e Tecnologia do Sudeste de Minas Gerais - Campus Rio Pomba, Rio Pomba – MG – Brazil.

[2]Universidade Federal Fluminense, Niterói - RJ – Brazil.

[3]Florida International University, Miami - Florida - USA

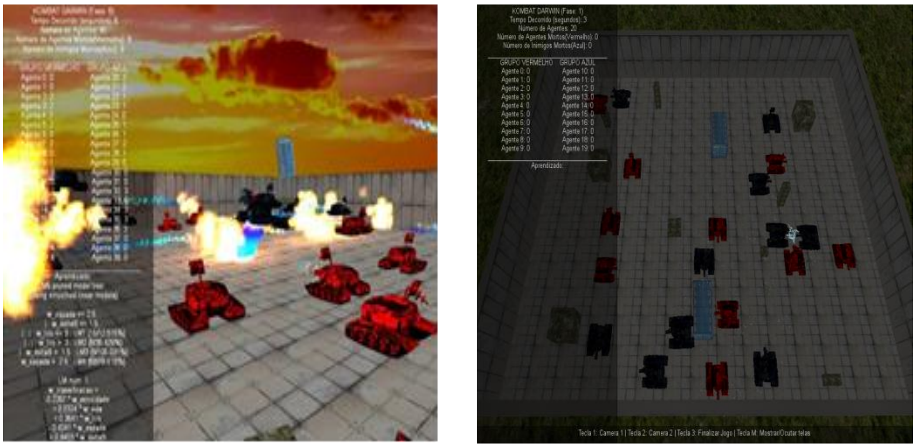

Fig.1. Darwin Kombat interface.

Abstract

The trend of traditional games is the gameplay even through the speed, frequency and other health conventional parameters. So it keeps players interested in the game. Another way is through machine learning techniques by adapting of the user.This paper proposes the use of an architecture based classifier MP5 and leveling techniques based on the top culling algorithm for ensuring dynamic balance. Through a strategy game, we concluded that this method works against the computer and against real players, then providing the fun factor.

Keywords: Leveled gameplay, machine learning, m5p, top culling.

Authors' contact:
{Alexcataguases}@hotmail.com,{crgdcomp,lindomar.lmp,endersoncsdep,heudesr}@gmail.com,esteban@ic.uff.br,troger001@fiu.edu.

1. Introduction

One of the concerns of developers of electronic games is the difficulty setting for the game tends to balance. This is made possible through the development of modern tools and techniques in Artificial Intelligence (AI).

According to Andrade et al. [2005], dynamic balancing of difficulty is divided into three basic rules: First, to adapt to the player's profile. Second, monitor the performance of the player. Third, keep the player interested. All this without the players realize that the system is making things simpler.

But according to [Spronck et al. 2004; Aponte et al. 2009; Andrade et al. 2005], in games with AI, the harder it more fun, it refers to experienced players, but beginners would not want to lose several times in a row.

According Spronck et al. [2004] in RPG games balancing techniques are used rule-based reinforcement learning. Our approach uses a game of strategy and reinforcement learning to generate rules. Resulting in greater realism.

Results presented in this article refer to those obtained by Andrade et al. [2005], distinguishing the following approaches:

- Experiments in a strategy game (Darwin Kombat) rather than fight.
- Use just online training instead of the hybrid.
- It includes experiments and conclusions involving computer versus human players.
- It was not used learning for balancing difficulty, but adaptation to the opponent.
- It is used rule induction (M5P) instead of selecting pre-established rules.

2. Difficulty Balancing Techniques

Many games provide values to determine the difficulty. This proposal is for beginners and experienced players can enjoy an appropriate challenge offered by the game. Typically, parameters such as strength and health influence the opponents, rarely tactic. Thus, even in difficulty level "hard" opponents have underperformed despite the strength and health is high.

According to Olesen et al. [2008] games can be balanced by the elastic factor, commonly used in racing games. That is, the AI increases to exceed the level of player, from this it will be reduced, giving balance to the game. However from the moment that the player realizes this effect the gameplay can be compromised and the game becomes predictable.

According [Hunckle et al. 2005; Chapman 2005; Andrade et al. 2006], the difficulty setting is often oriented Massively Multiplayer Online games, due to its size and attributes of gameplay. But this document shows that can also be applied to strategy games among other genres not only fighting game or RPG.

3. Increasing Difficulty through Machine Learning

Machine learning can be online and offline, the first is more efficient because the agent's behavior adapts to environmental changes. But, machine learning in games is rarely used due to its high risk. However, neural networks that are more difficult to implement, end up being more usual.

In the study by [Machado et al. 2010], has demonstrated the advantage of using numerical classifier algorithm (M5P) on qualitative attributes. When agents are successful, new attributes are generated from their base attributes, these new features will be used later.

Olesen et al. [2008] proved that the use of machine learning adapts the strategy and keep the dynamic balance in-game challenge. For games real time strategy (RTS), the methodologies used to generate neuro-evolution opponents. In this case, a training technique off-line (NEAT) is able to match the challenge of AI agent in relation to the skills of a player.

In contrast to Olesen et al. [2008], the hypothesis presented in this article, M5P, together with Top Culling the method can ensure the adaptation of the strategy of the player and the difficulty level through an online training.

4. Approach

In this section we describe the system with reinforcement learning algorithm M5P mediated top culling and the technologies needed for developing applications.

4.1 Architecture

The architecture of agents focuses on reinforcement learning, and is responsible for creating agents capable of better strategies. Based on [Machado et al. 2010; Machado et al. 2011] agents are classified without being generated by rules, its base is generated randomly, without training off-line, thus creating weak agents. So the result is sent to the server where the system creates rules that can classify a new agent based on their profile, generating a new stage/simulation.

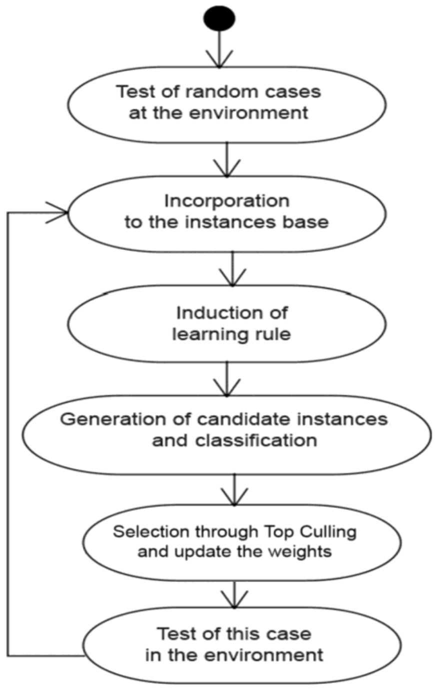

Fig.2. M5P-Reinforcement Learn Architecture with Top Culling module.

This architecture (Fig. 2) proposes to evaluate the rules generated from the "Top Culing" calibrating weights for the variables Wmin and Wmax. So the game continues, using the bases generated previously, applying the technical restrictions of Top Culling, restarting the cycle.

The value of Wmax is changed, with the intention to impose a greater challenge in the game. The value of Wmax is reduced by the constant Wdec (decrease) when the computer wins many times. And when it starts to lose Wmax is increased by Winc (increment).

Therefore, there is always a difference between Wmin and Wmax. Wmin works the same way, but the lower limits. Therefore, there is always a difference between Wmin and Wmax. To do this, will be diminished by Wmin Wdec when the range of weights between her and Wmax is very small.

Thus, strategies with a weight greater than and less than Wmax and Wmin not will be selected. Alternately, when the computer opponents lose many times the rules with high weights will be selectable (again), and opponents can use strong tactics to beat the opponents.

4.2 Technology

The technologies used are: 3D Unity (game engine) and Microsoft Visual Studio (IDE development server) with Weka, for the reasons described in [Machado et al. 2010]. The first performs simulation and the second learning process.

5. System Behavior Evaluation

To evaluate the effect of modified architecture, we use a modified version of a game classified as action-turn-based strategy called Darwin Kombat [Machado et al. 2010] (Fig.1). This trial program only the simulation of simple reactions and generation of agents with the learning process. This game was set to 20 players on each team (two teams, one for machine learning and one for the opponent).

The goal of each team is to eliminate all opponents. Each agent can have your profile created at the beginning of each match, changing the parameters of speed (motion), life (resistance shot), shot (ranged attack), shock (attack of contact) and / or delta-S (the space that is able to move before a turnaround at random), with values between 1 and 5. A game is defined as the set of 20 consecutive turns.

This game represents a prototype dynamic simulation system for the enemies and the environment and allows the generation of strategies for teams. The strategy of technical data of the agents is a feature also used in different commercial games.

By comparison, 20 tests were conducted in Darwin Kombat without using the technique of culling and top 20 tests using it. For graphics, the value in a column represents the number of enemy killed in each step, numbered below, the black lines show the amount of the death of machine learning agents (agent M5P), and the gray lines show the amount of static agents machine (single agent).

The sum of the attributes of the team M5P was set at 10 points and the opposing team by 15 points (with a profile correction) in both tests. Established to better characterize the learning phase of the algorithm, about three shifts (in both tests M5P algorithm starts at a disadvantage, changing progressively results). Based on the results, we can see that only using reinforcement learning, agents controlled by the machine, after the opponents win some battles, however, when using Top Culling, the results were satisfactory for this article, creating a balanced game.

No Top Culling Agent with learning gains for most of the fighting, but fighting with this technique is stable and allows alternating victories and defeats on all sides.

The emergency appeal filed by Machado et al. [2010] was maintained, as demonstrated in the following expressions generated in a test:

w_classification = (1)
0,3396 * w_life
- 0,5453 w_shock *
+ 2.5666

It demonstrates the tendency to generate agents with so much life and avoid agents with the attack of contact (shock).

6. Gameplay Evaluation

With both agents controlled by computer, where one was limited by Top Culling, tests were carried out with the players. Exactly 20 players in each game test. Players can change their strategy. At the end of each contest, the challenge for the agents with machine learning would learn to adapt to each player and the new strategy is limited by Top Culling (average results are shown in Fig.3 and fig.4).

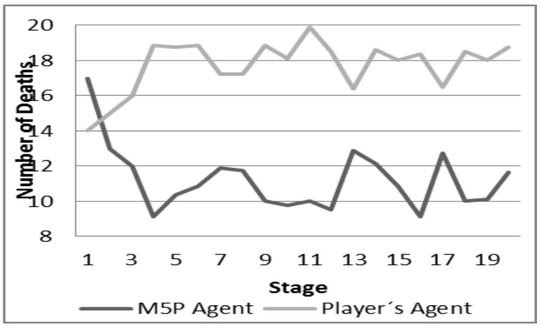

Fig.3. Graphic shows test without Top Culling Algorithm (versus player).

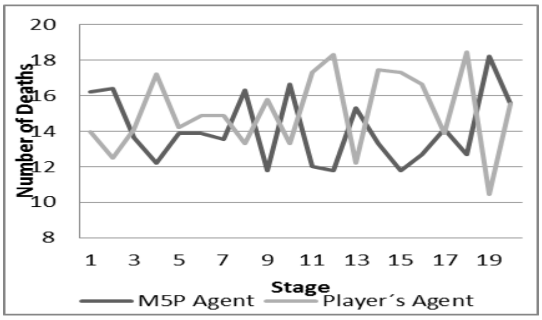

Fig.4. Graphic shows test with Top Culling Algorithm (versus player).

In assessing the conduct of the agent M5P against the player, we can deduce that even when a player changes the attribute values of the team, it makes no difference. Since each phase is analyzed by the agent, whether the attributes are changing or not.

We prepared a questionnaire with three questions, with answers valued from 1 to 5 (indicating intensity). From the average of the results, it can be said to play using Top Culling increases the level of challenge, without making the game too hard or too easy, indicating that the satisfaction of players increases. Moreover, it can be seen that users prefer to play with Top Culling just with learning, whereas the latter was more difficult to defeat(Fig.5).

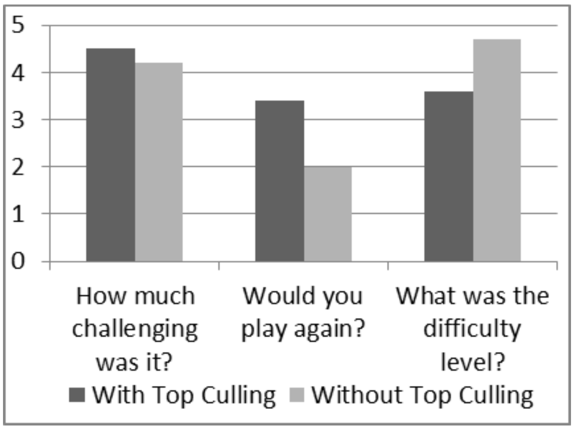

Fig.5. Graphic shows user satisfaction test.

7. Conclusion

This paper presents an alternative to the problem of difficulty in balancing dynamic games of strategy based on adapting user interaction.

Demonstrated that the use of an algorithm based architecture defined by the M5P - Top Culling technique can ensure the balanced development of the level of difficulty of a strategy game.

Through experiments on a turn-based game called Darwin Kombat, it was proved that the adaptation offered by the new architecture is able to limit the growth of learning and thus increase the funfactor game.

This work shows how the M5P algorithm transforms the game into a vicious circle, where agents adapt to the player and so the player changes his strategy, the agent adapts again, maintaining a cycle.

In future work, we intend to study and evaluate the gradual increase in difficulty to prevent the elastic factor, caused by the application of this technique without moderation.

Acknowledgements

This work has been partially funded by PET-MEC.

References

ANDRADE, G. CORRUBLE, V., RAMALHO, G. L.. Dynamic Game Balancing: An Evaluation of User Satisfaction. *In: Second Annual Artificial Intelligence and Interactive Digital Entertainment Conference*, 2006.

ANDRADE, Gustavo Danzi de; RAMALHO, G. L; SANTANA, Hugo Pimentel de; CORRUBLE, Vincent. Extending Reinforcement Learning to Provide Dynamic Game Balancing. *In: Workshop Reasoning, Representation, and Learning in Computer Games*, in IJCAI'05, 2005, Edinburgh.

ANDRADE G., RAMALHO G., SANTANA H., CORRUBLE V..“Automatic Computer Game Balancing: A Reinforcement Learning Approach” *In International Conference on Intelligent Agent Technology* , 2005

APONTE, M., LEVIEUX, G., & NATKIN, S., “Scaling the Level of Difficulty in Single Player Video Games”. *In Entertainment Computing–ICEC 2009*, 24-35. Springer.

HUNICKE R., CHAPMAN V. “AI for Dynamic Difficulty Adjustment in Games” *In ACM SIGCHI International Conference on Advances in computer entertainment technology*, 2005.

MACHADO, A. F. V., CLUA, E. W. and ZADROZNY B. “A Method for Generating Emergent Behaviors using Learning Machine to Strategy Games*” In SBGames.* 2010.

MACHADO, A., Carvalho, D., CLUA, E., DUARTE, C. G., MONTANARI, M. V. , REIS W. P. An Architecture Based on M5P Algorithm for Multiagent Systems. *ENIA.* 2011.

OLESEN, J., YANNAKAKIS, G., HALLAM, J.,. Real-time challenge balance in an RTS game using rtNEAT. *In: IEEE Symposium on Computational Intelligence and Games,* Perth, Australia. 2008.

SPRONCK P., SPRINKHUIZEN-KUYPER I., In., “Difficulty Scaling of Game AI” *In 5th International Conference on Intelligent Games and Simulation*, pp. 33-37, 2004.