Dynamic Difficulty Adjustment Through an Adaptive AI

Mirna Paula Silva, Victor do Nascimento Silva and Luiz Chaimowicz
Departamento de Ciência da Computação
Instituto de Ciências Exatas
Universidade Federal de Minas Gerais, Brazil
Email: {mirnasilva, vnsilva}@ufmg.br, chaimo@dcc.ufmg.br

***Abstract*—Dynamic Difficulty Adjustment (DDA) consists in an alternative to the static game balancing performed in game design. DDA is done during execution, tracking the player's performance and adjusting the game to present proper challenges to the player. This approach seems appropriate to increase the player entertainment, since it provides balanced challenges, avoiding boredom or frustration during the gameplay. This paper presents a mechanism to perform the dynamic difficulty adjustment during a game match. The idea is to dynamically change the game AI, adapting it to the player skills. We implemented three different AIs to match player behaviors: beginner, regular and experienced in the game *Defense of the Ancient (DotA)*, a modification (MOD) of the game Warcraft III. We performed a series of experiments and, after comparing all results, the presented mechanism was able to keep up with the player's abilities on 85% of all experiments. The remaining 15% failed to suit the player's need because the adjustment did not occur on the right moment.**

***Keywords*-Artificial Intelligence; Digital Games; Dynamic Difficulty Adjustment; Dynamic Difficulty Balance; Entertainment**

I. Introduction

The game industry is growing at a fast pace, globally generating more revenue than film and music industries [1]. Games are considered a great source of entertainment [2] and, due to that, the industry is increasingly investing more resources in research and development. This allows developers to create realistic graphics, deep narratives and complex artificial intelligence (AI), leading to games even closer to reality [3], [4].

Developing realistic games helps the improvement of players immersion which increases their satisfaction [5]. Although this is a good approach, it is not the only way to make games more attractive. According to Yannakakis [6], the player's psychological factor makes direct influence to this attractiveness, requiring the game to maintain the player interested on it. An approach to captivate the player into the game experience is to make the challenges directly associated to the player's skill [7]. However, a game may not suit the expectation of players with different skills. While a player may have a hard time in final levels of a game, there may be another player that cannot win the initial ones. This scenario requires that the game dynamically adjusts itself presenting challenges that suits the needs and skills of each player. This game adjustment can be performed by a technique called dynamic difficulty adjustment (or dynamic difficulty balancing).

This work aims to present a mechanism to perform the difficulty adjustment dynamically during a game match. To achieve this goal we observed and identified the behavior of three different types of player (beginner, regular and experienced) and developed an artificial intelligence that simulates each one of these. The main idea is to present an opponent that is challenging enough for the player without being too hard. Therefore we established an evaluation process to indicate moments during the game match where the player is increasing/decreasing his/her performance. A mechanism were developed to execute adjustments by changing the difficulty of the selected artificial intelligence and for each of these unbalanced moments, the mechanism analysis if is necessary to perform an adjustment or not. The framework selected to implement this approach is a modification (MOD) of the game WARCRAFT III, called DEFENSE OF THE ANCIENT (DotA). At this game, the player control one specific unit called hero and the main challenge is to destroy a main structure that belongs to enemy.

This paper is organized as follows: in Section 2 we present the related work and background on difficulty balance; Section 3 covers the game DotA used as framework of this work; Section 4 addresses the methodology and the proposed mechanism; Section 5 discusses the performed experiments and the obtained results; and finally, in Section 6 we offer our conclusions and future work.

II. Difficulty Balance

Difficulty balance, or difficulty adjustment, consists on doing modifications to parameters, scenarios and/or game behaviors in order to avoid the player's frustration when facing the game challenges [7], [8]. According to Mateas [9] and Hunicke [10], it is possible to adjust all game features using the correct algorithms, from storytelling to maps and level layouts, all online. These adjustments allow the game to adapt itself to each player, making he/she entertained throughout the game. To make this possible, Andrade et al. [11] describes that the dynamic difficulty adjustment must attend three basic requirements. First of all, the game must automatically identify the players' skills and adapt to it as fast as possible. Second, the game must track the

2159-6662/16 $31.00 © 2016 IEEE
DOI 10.1109/SBGames.2015.16

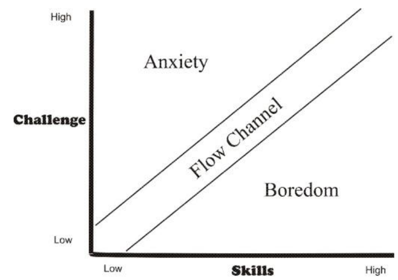
player's improvement and regressions, as the game must keep balance according to the player's skill. At last, the adaptive process must not be explicitly perceived by players, keeping game states coherent to previous ones. However, before applying the dynamic difficulty adjustment, it is necessary to understand the meaning of difficulty.

The meaning of difficulty is abstract in many ways and some aspects should be taken into account to evaluate and measure difficulty. For this measuring, we can consider level design characteristics [12], amount of resource or enemies [10], amount of victories or losses [13], among other metrics. Although, dynamic difficulty adjustment is not as simple as just giving player additional health items when in trouble. This problem requires estimation of time and intervention in the right moment, since maintaining the player entertained is a complex task in a interactive context [10].

A wide range of tasks and challenge levels can be found in games. For instance, tasks that require high skill and synchronism (First Person Games), tasks that require logic and problem solving skills (Puzzles), tasks related to planning (Strategy games), and so on [14]. According to Klimmt et al. [14], there is evidence that the completion of tasks and challenge overcoming are directly related to player satisfaction and fun. Yannakakis [15] developed a study about the most popular approaches for player modeling during interaction with entertainment systems. According to this study, most qualitative approaches proposed for player entertainment modeling tends to be based in conceptual definitions proposed by Malone [16] and Csikszentmihalyi [17].

Malone [16] defended the need for a specific motivation during gameplay to entertain the player. The necessary features to reach such motivation are: fantasy, control, challenges and curiosity. The use of fantasy as part of game world could improve player motivation, creating objects, scenarios or situations that the player could explore. Control is a player feeling that he/she is part of game control. Given the interaction of games, all of them makes the player feel involved in game control and the control levels can change from game to game. Challenge proposes that the game should pursue tasks and goals in an adequate level, making the player feel challenged to his/her limits. The uncertainty of completing tasks or goals provided by game mechanics encourages the player motivation. At last, curiosity suggests that game information must be complex and unknown, to encourage exploration and reorganization of information by players. Games must pursue parallel situations or scenarios from the main course since it helps to stimulate the player to explore the unknown [16], [18].

The qualitative approach proposed by Csikszentmihalyi [17] is called flow theory or flow model. According to the author, flow is a mental state that when the user is executing an activity in which he/she is immersed, feeling focused, completely involved and fulfilled during task execution. So, this model takes into account the psychological steps that players reach during gameplay. This occurs in an way that the main goal is controlling the challenge levels aiming to maintain the player inside the flow, avoiding to reach boredom (no challenges at all) or frustration (challenges are too hard). Figure 1 show a graph of flow theory presented by Csikszentmihalyi [17].



The Flow. After Mihaly Csikszentmihalyi, *The Flow (1990)*, p. 74

Figure 1. Demonstration of flow theory, by Csikszentmihalyi.

The model presented by Csikszentmihalyi shows how a task difficulty is directly related to the perception of who is executing it. The flow channel illustrates that difficulty can be progressively improved, since there exists time to the player to learn and improve his/her skills to overcome this challenge. Thereby, this model avoids frustration of very hard situations or boredom caused by very easy situations. Furthermore, Csikszentmihalyi and Nakamura [19] goes beyond and determine that the ratio of challenges to skills should be around 50/50 in order to produce enjoyable experiences.

On the other way, there are some studies that question how valid the ratio of challenges to skills really is as a measure of flow. Løvoll and Vittersø [20], for instance, presents a work with some empirical evidence that contest the idea that flow is produced when challenges and skills are harmonized. According to them, the interaction between challenges and skills as independent variables gave no support to the challenge skill ratio proposed by Csikszentmihalyi and Nakamura.

In a different approach, if we can balance the fantasy, control, challenge and curiosity proposed by Malone [16] and associate it to the progressive development of difficulty presented in flow model by Csikszentmihalyi [17], it is possible that the resulting game can entertain the player. However, using just these features does not show if game challenges are compatible with player skills. So, it is necessary measuring techniques to define when and how difficulty should be adjusted.

A. Evaluating the Difficulty Level

According to Andrade et al. [11], there are some different approaches to dynamically balance the difficulty level of a game. However, all of these approaches require measuring, implicitly or explicitly, the difficulty level that the player is facing on that moment. Defining player difficulty level is crucial to game mechanics evaluation and possible adjustments. This measurement can be done by using heuristics, for example the success rate of skill landing, the capture of enemy points, the time used to complete a task or any other metric that can evaluate the player. Missura and Gärtner [21] made a relation between game runtime, health and score in a way that it composes an evaluation criteria that performs the game difficulty adjustment. Demasi and Adriano [22] developed a heuristic function called "Challenge Function" that is responsible for describing the game state, and tries to show how hard the game is for the player in a given time.

Another way to track difficulty levels is using body language. Van Den Hoogen et al. [23] mentions that body language of a player could be related to his/her experience during play. According to the authors, there are evidences that show that specific postures, facial expressions, eye movements, stress over mouse/keyboard/joystick, and others, could evidence experiences like interest, excitement, frustration and boredom. For player experience evaluation, authors provide a monitoring ambient, in this place there where pressure sensors in the chairs and mouse. Also cameras were placed to register movements and facial expression. The results of this experiment show that the behaviors observed are directly related to the excitement level and dominance felt during the game. Nacke e Lindley [24], besides using cameras to capture body language, also used electrodes to track mental reaction from players during a First Person Shooter (FPS) match. The results obtained during player monitoring were based in flow theory proposed by Csikszentmihalyi [17], therefore, authors could observe if the players were inside the flow, anxious or bored during the gameplay.

Although the explicit measuring (external monitoring) of difficulty levels could provide fine results related to game adaptations to player's skill, it is impracticable to the dynamic difficulty adjustment. This can be observed because not all players have measuring tools at home and using such tools could be intrusive, since this could make the player uncomfortable by being monitored. Implicit approaches (metrics and heuristics) do not need external equipment and therefore these approaches are more popular among game developers. Besides, implicit approaches favors the conditions that players must not perceive that difficulty is being adjusted during gameplay.

This paper tries to perform a dynamic difficulty adjustment through the development of a mechanism that switches between three distinct artificial intelligences in order to provide an opponent that better suits the player's abilities. The mechanism perform several evaluations during the match indicating moments where the game is unbalanced and then execute the difficulty adjustment.

III. Defense of the Ancients

The game Defense of the Ancients (DotA) is a Multiplayer Online Battle Arena (MOBA) version of the game Warcraft III: Reign of Chaos and later to its expansion, Warcraft III: The Frozen Throne. The scenario objective is for each team to destroy the opponents' Ancient, a heavily guarded structure at opposing corners of the map. Players use powerful units known as heroes, and are assisted by allied heroes (played by others users) and AI-controlled fighters known as creeps. As in role-playing games, players level up their heroes and use gold to buy items and equipment during the mission.

According to Johnson et al. [25], MOBA games were found to offer less autonomy, more frustration and more challenges. These findings with respect to autonomy seems most likely to be a function of the fact that MOBA games involve fairly focused competition with other players. Moreover, the greater levels of frustration experienced may also be a function of the focused competition that occurs in MOBA games and the steep learning curve. With less focus on the immersive qualities of the game and greater focus on competing and cooperating with others, there is more potential for frustration with the performance of others players. This interpretation is supported by players reporting a greater challenge when playing MOBA games. Due to these characteristics, the use of a mechanism that performs the difficulty balance dynamically seems to be a viable alternative to minimize and/or avoid that such frustrations be experienced by the players. Therefore, the game Defense of the Ancients (DotA) was chosen to be the testbed of this work.

A. Gameplay

To provide challenges that suit the player's skills it is necessary to comprehend the gameplay that involves the game. The DotA game can be summarized into two teams playing against each other: the Sentinel and the Scourge. Players on the Sentinel team are based at the southwest corner of the map, and those on the Scourge team are based at the northeast corner. Each base is defended by towers and waves of units (creeps) which guard the main paths leading to their base. In the center of each base is the "Ancient", a building that must be destroyed to win the game.

Each player controls one hero, a powerful unit with unique abilities. In DotA, players on each side choose one of 110 heroes, each with different abilities and tactical advantages over other heroes. The scenario is highly team-oriented; it is difficult for one player to carry the team to victory alone. The Defense of the Ancients game allows up to ten players in a five-versus-five format.

Since the gameplay goes around strengthening individual heroes, it does not require focus on resource management and base-building, unlike most traditional real-time strategy games. When killing enemy units or neutral units, the player

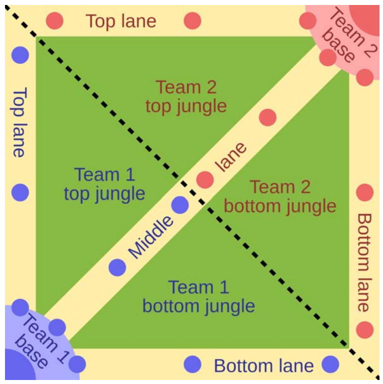


Figure 2. General map from MOBA games.

gains experience points and when enough experience is accumulated the player increases his/her level. Leveling up improves the hero's toughness and the damage it inflicts, and allows players to upgrade spells or skills. In addition to accumulating experience, players also manage a single resource of gold.

In DotA, besides the small periodic income, heroes can earn gold by killing hostile units, base structures, and enemy heroes. With gold, players can buy items to strengthen their hero and gain abilities. Also, certain items can be combined with recipes to create more powerful items. Buying items that suit one's hero is an important tactical element of the game.

The DotA game also offers a variety of game modes, selected by the game host at the beginning of the match. The game modes dictate the difficulty of the scenario, as well as whether people can choose their hero or are assigned one randomly. Many game modes can be combined, allowing more flexible options.

B. *Map*

The map is segmented into three different lanes, the top, the bottom, and the middle lane. Each one of these lanes leads to the other team's base, guarded by towers along the way. During the early laning phase of the game, most gameplay is centered around the lanes. Figure 2 represents a general MOBA map with its lanes, bases and towers along each lane.

The map area located between the lanes is called jungle. This is where neutral creeps can be found, which can be killed for gathering more gold and experience points. It is possible to level up by killing creeps in the jungle instead of in the lanes. This practice is called jungling.

Each team has defensive towers placed along the lanes leading to the Ancient. Towers inflict single target damage to heroes and creeps. In the early stages of the game, a hero can only take a few hits from a tower before dying, so one must be careful as to not get too close to towers until they have gained enough strength. In the Figure 2 the towers are represented by little circles placed in the lanes.

C. *Game Adaptations*

To use the game DEFENSE OF THE ANCIENTS as a testbed, some adaptations were made in order to better suit the needs of this work. As mentioned before, the original game allows the player to choose his/her hero among 110 different options. But, for this work, we chose to restrict this quantity to only 10 heroes, equally distributed between both teams.

Each hero has distinct characteristics, behaviors and abilities. Thereby, to better focus on the strategies and the development of abilities, we designed our artificial intelligence to control one specific hero. The selection performed were random and the chosen character is *Lion - The Demon Witch*. Given this choice, it became possible to classify which abilities and behaviors should be implemented so that the artificial intelligence would work with a consistent behavior during the game match.

IV. METHODOLOGY

Our difficulty adjustment mechanism consists in the development of three different types of artificial intelligence that will be chosen during the match in order to present challenges that suit the player's skills. To select the right opponent, a difficulty evaluation is perform during the game and if it indicates that the players are not evolving in the same pace, it executes the necessary adjustment. Throughout this section, we shall address the artificial intelligence developed, the game features, the difficulty evaluation process and the mechanism to dynamically adjust the presented difficulty during a match.

A. *Artificial Intelligence*

To be able to provide an opponent that can face different skilled players, the artificial intelligence must be implemented with distinct ability levels to simulate the most different behaviors played. Since the artificial intelligence must simulate an opponent player, the developed algorithm implements actions and behaviors to a hero unit. During a game match this hero should follow the player's performance, so if the player is having a good evolution, the hero controlled by artificial intelligence must be able to also do the same. However, if the player is not evolving enough or if his/her development started to decrease, the artificial intelligence hero must also decrease its strategies and skills and keep up with its opponent.

The hero behavior was divided into three categories: easy mode, regular mode and hard mode. Each one of these categories has singular aspects that aims to be suitable to players with different abilities. These modes will be described as follows.

Easy Mode:: In the easy mode, the hero performs regular attacks every time an enemy enters in its attack range. When an allied tower is under attack, the hero detect the need for defense and moves towards the attacked ally in order to defend it. Another strategic action is how the hero chooses the enemy tower to be its main target. Every time the hero starts a moving action, it analyses which one of the enemy's tower has taken more damage and is closer to be defeated. Once it finds, the hero sets that tower as the main target and go in that direction. It is important to mention that, in the easy mode, all the attacking actions that the hero performs are basic attacks. The hero also retreats as a defense strategy. So when its health points are below 30%, it starts to retreat towards the allies base, where it can recover its health when it reaches a specific recovery building. The easy mode was created for beginners or some less skilled players, where the implemented strategies are not very complex and does not use any special character skill (also known as spells).

Regular Mode:: In the regular mode, besides the strategies implemented for the easy mode, the hero also starts to manipulate items. The item manipulation is very helpful to improve the hero's attributes and also to recover some attributes that has been decreased, for example, items to recover health points or mana. Likewise, there are items to increase attributes like strength, speed, intelligence, among others. As part of the defense strategy, if the hero's health points reach 30% or less, it will first use some health potion to recover it and if these items are over, then the hero starts to retreat towards its base. The regular mode was created to cover those players that have already some experience and know how to use some of the game functionalities in his/her favor but are not experts yet.

Hard Mode:: The hard mode has all the strategies implemented on both preceding modes, besides its own specific actions. Here, the hero goes beyond item manipulation and starts to learn, improve and cast spells. Spells are unique skills that each hero has. These spells can give a more effective damage on the enemy, can boost the recovery of its own attributes (like mana or health points), can give some kind of advantage to allied units (like freezing the enemies), among other possibilities. Every time the hero gains a new level it also gains one attribute point to distribute among its spells. So in this mode, besides the regular attack, the hero also casts spells to attack enemies or defend allies. Here we also decide to implement a new strategy for a head-to-head combat. In order to avoid losing the combat against another hero, the artificial intelligence algorithm keeps monitoring the area around its hero. Therefore, if an enemy hero enters the monitored area, the hero controlled by the artificial intelligence will take advantage on that and will begin to attack it. The strategies to defense allied towers and to retreat are the same developed on regular mode. The hard mode was created to cover those players that have more experience on the DotA game and also know how to use the game functionalities in their favor. This kind of player may be an expert on the game or a quick learner.

		Artificial Intelligence		
		Easy Mode	Regular Mode	Hard Mode
Defense Strategy	Defend Allied Towers	X	X	X
	Retreat	X	X	X
	Item Manipulation		X	X
Attack Strategy	Main Attack	X	X	X
	Target Selection	X	X	X
	Track Enemy Hero			X
	Cast Spell			X

Figure 3. Summary of the developed strategies for each artificial intelligence.

The table displayed in Figure 3 summarizes all the developed difficulty modes and their strategies.

B. Difficulty Evaluation Process

A difficulty evaluation process was elaborated to be performed during the game and indicate when the players are not evolving in the same pace. For that, it was necessary to observe which game features should be analyzed and how to properly use the information from each one of them. The analyzed features and the evaluation process will be described below.

1) Game Features: To evaluate a game match it is crucial to identify which features can represent the players' performance and should be considered relevant to the evaluation. In our testbed, we identified three important features that can illustrate the player's behavior during a DotA match. These features are: Hero's Level, Hero's Death and Towers Destroyed. Each one of these features will be described below:

Hero's Level:: This feature represents the player's evolution during a match, where the greater is the level value, the stronger is the character. Although this feature represents the evolution, it should not be the only analyzed feature because it is possible that the player increases his/her hero's level without really increasing his/her abilities. For example, the player can keep the hero closer to battles without engaging in any fight and, by doing that, it will gain some experience points that are shared among the allies that are closer to the battle and it will help the hero to evolve its level. Thereby, even if all players have heroes with equivalent levels, this feature alone does not give a real track on the game balance.

Hero's Death:: This feature is responsible for showing how many times the hero has died during a DotA match. Differently from all other features, the hero's death may represent the player's performance and the level of difficulty that he/she is facing more accurately. For example, an inexperienced player, even having a hero with a high level, may have a high death rate, since he/she may not know how to use more properly the characteristics and peculiarities of his/her character as well as a possible lack of game strategies. Thereby, this feature seems to represent more accurately how well the player is facing the game challenges.

Towers Destroyed:: This feature is the amount of enemy's towers destroyed by the allied team. It represents the team expansion and dominance over the map. Although this feature is not directly related to the player's performance, since other allies can also destroy towers, it gives us a good notion of the game's progress and team expansion over the map. Therefore, if a team is quickly progressing over the map, it may represent that the game is unbalanced.

2) Difficulty Evaluation Process: In order to perform a dynamic difficulty adjustment, it is necessary to evaluate the game from time to time and verify if the game is presenting challenges suitable to the player's performance. If the player is having a poor performance, the game should be capable to identify that and reduce its difficulty. In the same way, if the player evolves faster than the challenges presented, the game should increase its difficulty.

Once we have defined the game features that must be analyzed, this process can be summarized into a creation of an heuristic function that will keep track on the player's performance and inform when it is necessary to adjust the difficulty. This heuristic function will be our evaluation method during the game match and from now on it will be called as evaluation function. So, considering the features mentioned before and the impact that each one represents on the player's performance, we have:

$$P(x_t) = H_l - H_d + T_d, \tag{1}$$

where $P(x_t)$ is the performance function of player x on time t. H_l is the hero's level, H_d is the hero's death and T_d is the towers destroyed. It is important to mention that the values of this features are related to the player and his/her hero. Once you have computed the performance value of the player from start time ($t = 0$) to current time ($t = i$), it is necessary to do the same for time $t = i - 1$. After having both values, it is possible to calculate the current evolution of the player, as shown in the equation below:

$$P'(x) = P(x_t) - P(x_{t-1}). \tag{2}$$

Once the performance function was calculated for both players (x and y) the evaluation value can be obtained by:

$$\alpha = P'(x) - P'(y), \tag{3}$$

where α is the difference among performances. It is important to mention that player x is the one that we are analyzing and player y is the one controlled by the artificial intelligence system. Therefore, the player y is the one that will have its difficulty adjusted during the game.

C. Dynamic Difficulty Adjustment Mechanism

The proposed mechanism is the key to make the adjustment work properly during the game. Until now we have only showed how to verify if the player's performance is balanced to a certain opponent or not. Thereby, the main task of the implemented mechanism is to analyze the α value and perform or not the difficulty adjustment at the game time t.

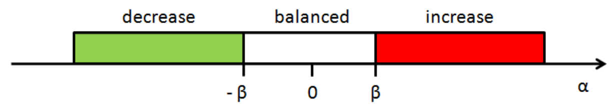


Figure 4. The verification performed by the adjustment mechanism.

The mechanism works by evaluating the α variable and constantly verifying if this variable is within the β's range. Where β represents the limit value of the evaluation function. This value means how far a player can perform better than the other player, without considering the game unbalanced. If the value of $|\beta|$ is a huge number, then the adjustment will occur with less frequency, since it may take some time to α overcome β. Likewise, if $|\beta|$ is a small number, then the adjustment will occur more frequently, since it may overcome β more easily. And if α stays inside the limits values of $-\beta$ and β, it means that both players are having a similar performance and therefore, the match is currently balanced. The Figure 4 illustrates this approach.

V. EXPERIMENTS

In order to verify the effectiveness of the proposed mechanism, a series of experiments was performed. The players' performance were analyzed along with the behavior of their heroes. The dynamic adjustment mechanism was also observed, as well as its variations and the impact caused on the matches.

On each experiment we ran the game with the static artificial intelligence controlling one team against the dynamic artificial intelligence controlling the other one. Was performed a game set with 20 matches for each experiment and after observing the results contained in the gamelog of several matches we defined that the β limits should be -1 and 1. Therefore, every time the difference among performances (α) exceeds the β limits, the difficulty of the dynamic AI should be modified accordingly to the obtained value.

A. Baseline

First, we performed an unbalanced match in order to stipulate a baseline to compare with the obtained results from all three experiments. This baseline match is set by two different artificial intelligence players with static behavior. One of them is on easy mode, representing a player without experience, and the second one is on hard mode, representing a very experienced player. The results of the mentioned match are shown in Figure 5, where the difference among both performances can be noticed.

The player's performance is measured taking into account his/her current state during the match. The positive peaks

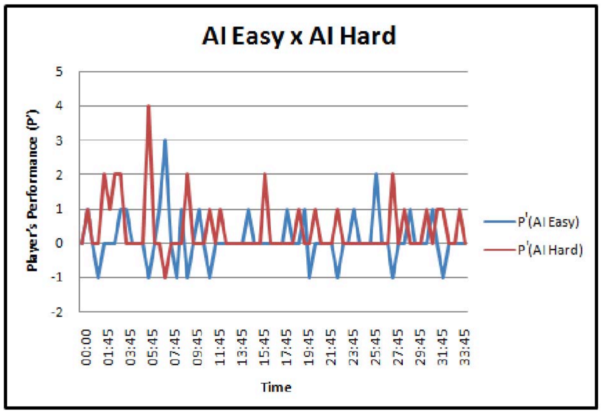


Figure 5. Graphic with baseline values obtained from a match with static difficulty

represents moments where the player improved his/her abilities when compared to his/her last state. Likewise, negative peaks means that the player had his/her performance decreased based on his/her last game state. Converting these to game situations, when a hero gains a level or the team manage to destroy a tower, then this will impact positively in his/her development, increasing the player's current performance. Similarly, if a hero dies this will result in a negative impact in his/her development decreasing the player's current performance.

During this game match, the hard mode player kept increasing his performance, presenting only one time of regression in his development. Meanwhile, the easy mode player performance was very unstable, presenting many moments of regression in his development. Therefore, we can consider that a match will be balanced if the difference among both performances were not divergent. So, examining once again the graphic, it is possible to observe that each performance peak shows itself as an appropriate moment to execute a difficulty balance in order to get the players' performance closer to each other.

Figure 6 shows the accumulative performance value from each player during this particular match. On this graph, it becomes clear that the hard mode player evolves much faster than the easy mode player. This greater performance evolution can be related to the fact that the hero increases his level rapidly and has a low amount of fatalities. Differently from the easy player that although his hero had a great level development, the amount of deaths was also high, leading to a poor performance when compared to the hard mode player. Therefore, due to that difference between them, the adjustment appears to be necessary in order to minimize this disparity among their behaviors and present a more fair and competitive game.

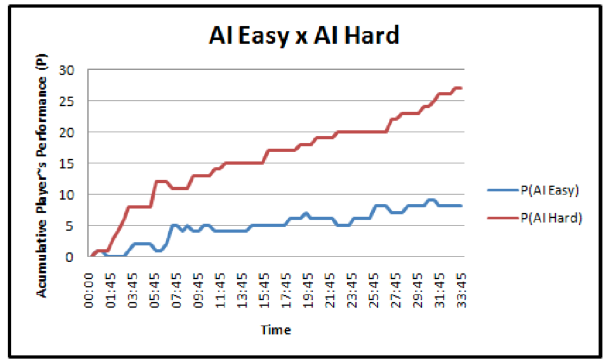


Figure 6. Graphic with the accumulative player's performance (P) value.

B. Easy x Adaptative

The experiments were performed using the artificial intelligence developed to control the heroes, one from each team. For player A we decided to use a static artificial intelligence in order to simulate the behavior of a human player. For player B we applied the proposed mechanism, where this player should keep its abilities suitable to player A and for that it should perform a dynamic difficulty adjustment. The first set of experiments, we manage to simulate a beginner player with player A. The player B started on regular mode and during the match it should be balanced to better fit the skills of player A. Figure 7 shows the performance of both players during this match (P'), while Figure 8 shows the results of the evaluation function (α) and the difficulty adjustments made during the game.

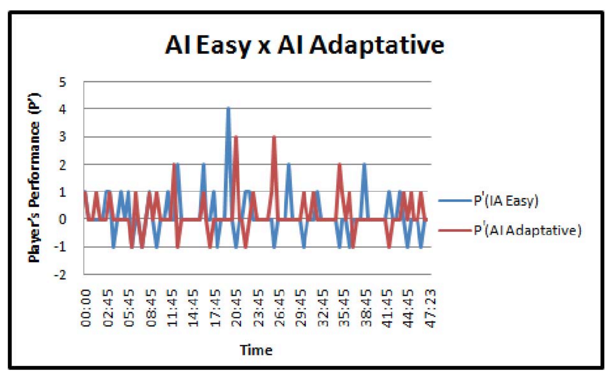


Figure 7. Graphic with players performances during one match.

As mentioned before, the player's performance is measured by taking into account his/her current state during the match. The positive peaks represents moments where the player had improved and negative peaks means that the player had decreased based on his/her last game state. Figure 7 shows that the adaptative artificial intelligence (player B) managed to keep its performance similar to its opponent, the easy mode player A.

On Figure 8 we can track how well the adaptative player

(player B) manage to be compatible with player A during the match. When the evaluation function shows negative peaks, it means that the difficulty should be adjusted and decreased by one level. Likewise, if there are positive peaks resulted by the evaluation process, than the difficulty of the adaptative player should be increased by one level. Moments where the evaluation function remains constant (equals 0) means that the performance of both players are very similar and due to that no adjustment is necessary at this time. Therefore, the difficulty can be maintained.

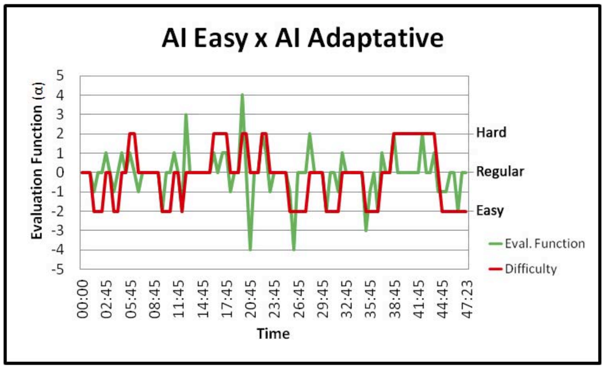


Figure 8. Graphic with the difficulty adjustments performed by the mechanism during one match.

It is important to mention that the difficulty adjustment is performed by increasing or decreasing one level of each time. With this approach we minimize the possibility of the opponent player noticing the behavior change. After analyzing this set of experiments and study the gamelogs obtained from each one, we observed that in 85% of the matches, the adaptative player B managed to keep the game balanced and as result of each match, player A won 60% of the matches and player B won 40%.

C. Regular x Adaptative

On the second set of experiments, we kept using the artificial intelligence developed to control two players, one from each team. Here, we manage to simulate an intermediary player with player A using a static artificial intelligence on regular mode. For player B we applied the proposed mechanism, starting it on regular mode and during the match it should keep the game balanced. Figure 9 shows the performance of both players (P') during one single match. Likewise, Figure 10 shows the results of the evaluation function (α) during the game and the difficulty adjustments made over the match.

The analysis performed in this set of experiments is pretty similar to the previous one. The positive peaks represents moments where the player had improved and negative peaks means that the player had decreased its performance. In Figure 9 we can observe that the adaptative artificial intelligence (player B) tried to follow its opponent's performance (player A) presenting similar peaks at the same time, or in some moment closer, to its opponent.

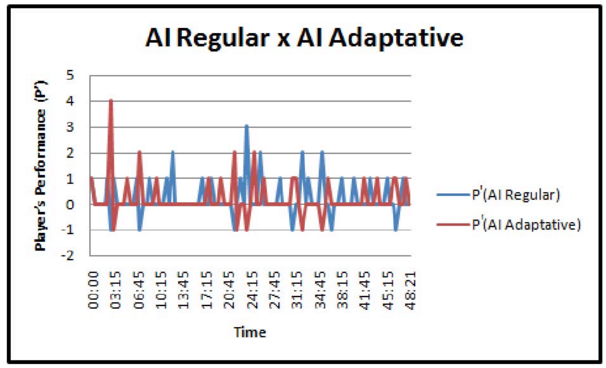


Figure 9. Graphic with players performances during one match.

On Figure 10 we can follow all the adjustments made during the match. The adaptative player spent most of its time alternating between the regular mode and the hard mode. This variation can be understood as moments where the player B were having a poor development when compared to player A, and the need of increasing the difficulty was perceived. Similarly, when player's B behavior were standing out the need for reducing the difficulty could also be seen. The graphic also shows that player B stayed balanced during the game. Furthermore, after analyzing this second set of experiments and study all gamelogs collected, we observed that the players had a compatible performance in 90% of the matches. The results of the matches can be splitted into a 50-50 victories.

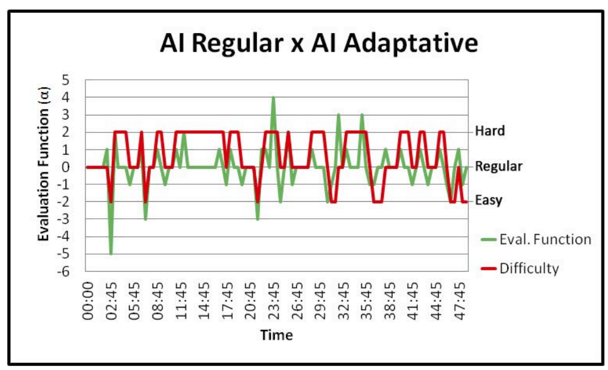


Figure 10. Graphic with the difficulty adjustments performed by the mechanism during one match.

D. Hard x Adaptative

On the last set of experiments, we manage to simulate an expert player (player A) against the developed adaptative player (player B). As we mentioned before, the adaptative player started on regular mode and changed its behavior during the match in order to keep the game balanced.

Figure 11 shows the performance (P') of both players during one match. Likewise, Figure 12 shows the results of the evaluation function (α) during the game and the difficulty adjustments made over the match.

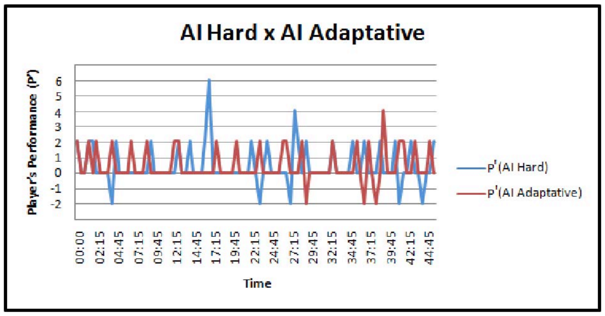


Figure 11. Graphic with players performances during one match.

Analyzing the results from Figure 11, the adaptative player started developing a better performance than player A in the beginning of the match. Therefore, it was detected that the difficulty should be reduced in order to keep the balance (Figure 12). After that, they keep their performances very close and the difficulty keeps alternating between easy mode and regular mode until player A can present himself/herself better/stronger than player B. The opposite can also be seen, when player B keeps alternating between regular mode and hard mode in order to reach player's A performance.

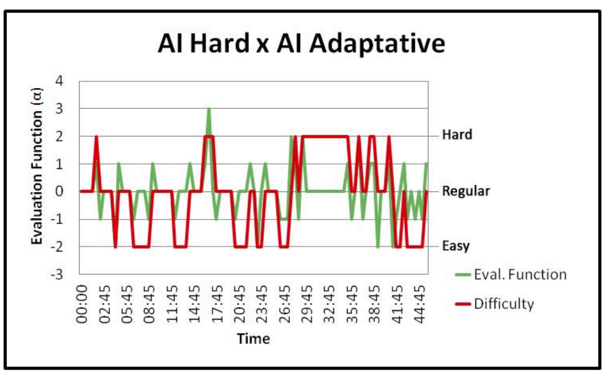


Figure 12. Graphic with the difficulty adjustments performed by the mechanism during one match.

Furthermore, after analyzing the gamelogs collected from each game, we observed that the adaptative player (player B) changed its difficulty and succeed to keep the match balanced on 80% of the experiments. As result of the battles, player A won 45% of the matches.

E. Discussion

Not all the cases presented the expected results, which has resulted in unbalanced matches. To get to this conclusion, we observed all the executed matches and studied all the collected gamelogs. These gamelogs kept track of the game on every 15 seconds, recording the current situation of both teams, the related features, their values, among other information. Once the game was finished, we started to translate those collected information, comparing the values from both heroes and making the necessary assumptions.

Considering all the performed experiments 10% of it were unbalanced because the mechanism took too long to perform each adjustment, leading to a great difference between the players performance. So, when the players were getting closer to a balance, the match has ended. On the other way, 5% of the executed experiments were unbalanced due to an excess of adjustments. In these scenarios, the adjustments were being performed too quickly, leading player B to not evolve properly during the match, which resulted an easy game for player A.

	AI Adaptative	
	Win	Lose
AI Easy	8	12
AI Regular	10	10
AI Hard	11	9

Figure 13. Final results from all matches performed on the experiments.

After performing all the experiments it was possible to summarize the obtained final results from the game matches. Figure 13 shows the amount of victories and losses of the adaptative AI against the easy, regular and hard modes. According to these values, we can observe that the game kept impartial once both players had very close results, showing that exist the possibility of the human player win or lose the match, it will relay on his/her abilities.

VI. Conclusion

The dynamic difficulty adjustment consists in an alternative towards the definition of the game challenge levels. This adjustment is dynamically performed, making it possible to track the player's skills and adjust itself during game runtime.

The presented work aimed to increase the player's entertainment by providing a mechanism that adjusts the game AI according to the player's skills. This mechanism was implemented on a game modification of WARCRAFT III, called DEFENSE OF THE ANCIENT (DotA). After performing experiments that simulate the three main player's behaviors (beginner, regular and experienced), it was possible to verify that the dynamic difficulty adjustment mechanism was able to keep up with the player's abilities on 85% of all experiments. On the remaining experiments that failed to suit the player's skill, 10% of it occurred because the adjustment mechanism spent too much time to perform each needed adjustment which leaded to a great difference between the players performance. And the last 5% of it occurred due to an excess of adjustments that was performed too quickly, without giving enough time to the game to evolve properly.

Given the presented results, we can conclude that the proposed mechanism behaved as expected and is capable to offer a game match compatible with the simulated player's performance. Also, after observing all obtained results, we can state that the key to a balanced game is to keep changing the difficulty of the adaptative player in order to follow the performance of the human player and avoid boredom and frustration.

As future work, the dynamic difficulty adjustment mechanism will be improved in order to decrease the amount of cases where the balance did not worked properly. We also intend to perform some qualitative experiments on human players with different experiences on the game DEFENSE OF THE ANCIENTS (DotA) to better evaluate the developed mechanism.

REFERENCES

[1] P. Thompson, R. Parker, and S. Cox, "Interrogating creative theory and creative work: Inside the games studio," *Sociology*, p. 0038038514565836, 2015.

[2] A. Nareyek, "AI in computer games," *Queue*, vol. 1, no. 10, p. 58, 2004.

[3] M. C. Machado, E. P. Fantini, and L. Chaimowicz, "Player modeling: Towards a common taxonomy," in *Computer Games (CGAMES), 2011 16th International Conference on*. IEEE, 2011, pp. 50–57.

[4] A. M. Smith, C. Lewis, K. Hullett, G. Smith, and A. Sullivan, "An inclusive taxonomy of player modeling," *University of California, Santa Cruz, Tech. Rep. UCSC-SOE-11-13*, 2011.

[5] D. A. Bowman and R. P. McMahan, "Virtual reality: how much immersion is enough?" *Computer*, vol. 40, no. 7, pp. 36–43, 2007.

[6] G. N. Yannakakis and J. Hallam, "Towards optimizing entertainment in computer games," *APPLIED ARTIFICIAL INTELLIGENCE*, 2007.

[7] B. B. P. L. de Araujo and B. Feijó, "Evaluating dynamic difficulty adaptivity in shoot'em up games," in *Proceedings of the XII Brazilian Symposium on Games and Digital Entertainment - SBGames 2013*, São Paulo, Brazil, Oct. 2013, pp. 229 – 238.

[8] R. Koster, *Theory of fun for game design*. O'Reilly Media, Inc., 2010.

[9] M. Mateas, "Interactive drama, art and artificial intelligence," 2002.

[10] R. Hunicke, "The case for dynamic difficulty adjustment in games," in *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology*. ACM, 2005, pp. 429–433.

[11] G. Andrade, G. Ramalho, H. Santana, and V. Corruble, "Extending reinforcement learning to provide dynamic game balancing," in *Proceedings of the Workshop on Reasoning, Representation, and Learning in Computer Games, 19th International Joint Conference on Artificial Intelligence (IJCAI)*, 2005, pp. 7–12.

[12] R. A. Bartle, *Designing virtual worlds*. New Riders, 2004.

[13] G. Xavier, "A condição eletrolúdica: Cultura visual nos jogos eletrônicos," *Teresópolis: Novas ideias*, 2010.

[14] C. Klimmt, C. Blake, D. Hefner, P. Vorderer, and C. Roth, "Player performance, satisfaction, and video game enjoyment," in *Entertainment Computing–ICEC 2009*. Springer, 2009, pp. 1–12.

[15] G. N. Yannakakis, "How to model and augment player satisfaction: a review," in *Proceedings of the 1st Workshop on Child, Computer and Interaction (ICMI'08), ACM Press, Montreal, Canada*, 2008.

[16] T. W. Malone, "Toward a theory of intrinsically motivating instruction*," *Cognitive science*, vol. 5, no. 4, pp. 333–369, 1981.

[17] M. Csikszentmihalyi, *Flow*. HarperCollins, 1991.

[18] S. Egenfeldt-Nielsen, J. H. Smith, and S. P. Tosca, *Understanding video games: The essential introduction*. Routledge, 2013.

[19] M. Csikszentmihalyi and J. Nakamura, "Effortless attention in everyday life: A systematic phenomenology," 2010.

[20] H. S. Løvoll and J. Vittersø, "Can balance be boring? a critique of the challenges should match skills hypotheses in flow theory," *Social indicators research*, vol. 115, no. 1, pp. 117–136, 2014.

[21] O. Missura and T. Gärtner, "Player modeling for intelligent difficulty adjustment," in *Discovery Science*. Springer, 2009, pp. 197–211.

[22] P. Demasi and J. d. O. Adriano, "On-line coevolution for action games," *International Journal of Intelligent Games & Simulation*, vol. 2, no. 2, 2003.

[23] W. Van Den Hoogen, W. Ijsselsteijn, and Y. de Kort, "Exploring behavioral expressions of player experience in digital games," in *Proceedings of the workshop on Facial and Bodily Expression for Control and Adaptation of Games ECAG 2008*, 2008, pp. 11–19.

[24] L. Nacke and C. A. Lindley, "Flow and immersion in first-person shooters: measuring the player's gameplay experience," in *Proceedings of the 2008 Conference on Future Play: Research, Play, Share*. ACM, 2008, pp. 81–88.

[25] D. Johnson, L. E. Nacke, and P. Wyeth, "All about that base: differing player experiences in video game genres and the unique case of moba games," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 2015, pp. 2265–2274.