

Techniques for Dynamic Adaptation of Game AI for Enhancing User Experience

Nakul S. Joshi
Information Technology, Sinhgad
Institute of Technology and
Science, Pune, India

Ajinkya M. Vaidya
Information Technology, Pune
Vidyarthi Griha's COET
Pune, India

Unmesh V. Deshmukh
Information Technology, Pune
Vidyarthi Griha's COET
Pune, India

ABSTRACT

No matter how advance the game AI is, it falls victim to repetitive behavior which after certain interval starts representing a pattern in the game which the user can exploit frequently. The elimination of this loophole will not only help the game AI to better counter the user behavior but also help it to learn and improve itself. This will enhance user experience and encourage him to think and act in a different manner while keeping the difficulty level same. As the modern day games strive to perfectly emulate human behavior in their AI, they still fall short of embedding human intelligence and are unable to inject their AI with the aspect of 'rational' thinking. The field of AI has seen remarkable growth and the advent of very sophisticated and accurate techniques to model user behavior which can help the field of game development to finally achieve a seamless transition from satisfactory to a prolonged exciting experience. This paper analyses the need to overcome the above mentioned issues and discusses the techniques to do so. The algorithms proposed in this paper achieve a common goal taking different approaches; these are DDA (Dynamic Difficulty Adjustment) for FPS games and DADA (Dynamic Adaptability using Data Analysis) for others. These techniques incorporate dynamic learning, elevating the games to become ever challenging and highly enjoyable instead of being monotonous.

General Terms

Game AI, Dynamic Learning

Keywords

DDA, FPS, DADA

1. INTRODUCTION

Modern day gaming has evolved to a whole new level where the graphics have gone from strength to strength from their predecessors. After a brief overlook into the history of game development the following three stages can be ascertained-

1.1 Stage I

In early days, the game development was restricted to a sub-par level as the processing power of processors then was not as adequate as it is now. The games relied on specific patterns and had lot of limitations which led to the user experience being not up to the mark. We can cite the examples of classical games like Donkey Kong, Galaga, etc. where the coders relied on prescribed patterns or restrictive motions for enemies. The users had a limited gaming experience and could figure out the fallacies in it.

1.2 Stage II

This stage witnessed massive growth in the field of game development. The graphics improved markedly from the previous stage. The games were characterized by drastic improvement in user experience which propelled the industry to

new heights. The combination of improved graphics and enhanced AI together marked a very important phase in the history of game development and attracted potential investors which led to the invention of specialized graphics processors. These processors shifted the focus of developers to develop games with sophisticated graphics.

1.3 Stage III

This stage represents the current modern day games. The norm for graphics nowadays is very high which has led to the developers concentrating on it more than other factors. The selling point of modern day games is slowly but surely changing from the complex and intricate AI to stunning and extensive graphics and sophisticated storylines. This has stymied the growth of AI and the importance of dynamic learning in games is being overlooked.

The field of AI has witnessed gigantic growth. The AI is increasingly leading towards "real" intelligence techniques (as defined by academic AI) instead of the old standby of pre-scripted patterns or behaviors that only mimic intelligent behavior. Instead of creating a set-in-stone list of AI behaviors and reactions to human actions, we can keep the behavior mix exhibited by the actual human reactions. While the graphics remain an integral part nothing dampens the user experience like static and non-adaptable AI. [1]

In this paper we state, DDA (Dynamic Difficulty Adjustment), which can be used in only specific genres of games like FPS that can make the AI more adaptable and dynamic. At the same time, the proposed algorithm i.e. DADA can specially be used for the wide variety of games in which the solution of making the AI more adaptable to different user keeping the difficulty level of the AI same can be achieved.

2. EXISTING AI ISSUES

The overall game development can be categorized into two main components- graphics and Game play. The game play mainly constitutes of specific patterns and techniques related to AI. The game play has improved remarkably in modern day games and yet expert and novice users alike can recognize the flaws in the AI far too easily. While the AI certainly manages to engage the users in the game yet it fails to adapt to each individual user leading them to believe that this complex AI is instead a prescribed pattern. This point can further be illustrated with an example of the popular football game viz. the FIFA series [2]. The AI here is set up in reference to the pre-set difficulty level chosen by the user. The main issue most of the users face here is transition from one level to another. As it takes too much time to adapt to the new and the higher difficulty level, somewhere down the line the user loses his interest and switches back to lower difficulty. While making the games competitive must

certainly be on the minds of a game developer, one key fact must always be followed-“The user does not like to lose repeatedly”. Since the user is already accustomed to lower difficulty level he now finds this far too easy. While in the first scenario the game AI was fulfilling its purpose of engaging a player in a grueling, now the user is led to believe that AI is letting the user win far too easily. The solution to this problem can be cited from real world example of football. The manager of a team changes the tactics of a team according to the opposition. Maybe a manager might not change his tactics prior to the game but facing a dire situation he might rectify it. While the game AI succeeds in mapping most of the human behavior the above situation is completely ignored which leads to a user always finding weakness in AI or a successful pattern to win. The user can repeat his behavior, that he has found to be successful and it works most of the time making the AI redundant. One of the biggest limitations of AI is that it fails to emulate human intelligence; while a normal human player is susceptible to occasional mistakes he is still capable of producing some inspired moments. This experience is observed in multiplayer gaming as the user is tested against his mettle and has to be on his toes all the time. Somewhere down the line an average user still prefers the option of playing against a human entity rather than the computer bots highlighting a major flaw in the approach for the development of modern day AI. The AI in the FPS (First Person Shooting) games is limited as the enemies are mostly stationed at same point and exhibit the same behavior every time the player dies and restarts. There are certain different ideologies while playing a FPS game, some users prefer to play offensive and others defensive. The AI needs to adapt to this behavior and devise strategies according to the overall game play of the user making him think that he is playing a truly intelligent entity which is able to analyze him and trace his faults.

The Figure 1 depicts an average scenario for everyday games where a user excitement remains static after a period of time. Also the learning curve states that it declines gradually w.r.t time.

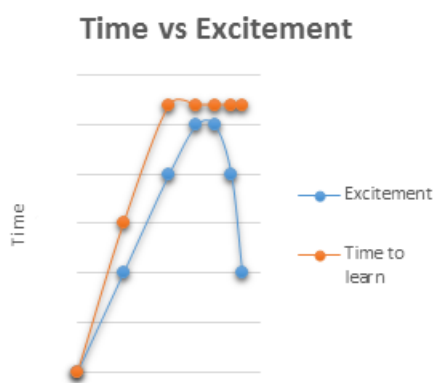


Figure 1. Time vs level of Excitement Graph

3. DYNAMIC DIFFICULTY BALANCING

Dynamic game difficulty balancing, also known as dynamic difficulty adjustment (DDA) is the process of automatically changing parameters, scenarios, and behaviors in a video game in real time. The goal of dynamic difficulty balancing is to keep the user interested from the beginning to the end and to provide a good level of challenge for the user. DDA mechanism attempts to change the difficulty according to the user in real time instead of presetting the difficulty. It is a counter mechanism where the AI reacts to the way the user plays. [3]

A simple DDA algorithm consists of the following two steps-

3.1 Step 1:

Measure the difficulty the user is facing at that given instance.

This measure can be performed by a heuristic function. The heuristics could be any information that will help us understand the current situation of the user. An example could be the time required to complete a task. This heuristics are completely dependent on the genre and may differ from one game to another.

3.2 Step 2:

Control the game environment settings in order to make the user challenges easier or harder.

After a complete analysis of the current user scenario we manipulate the game environment to suit his condition at that instance. This keeps the user interested as the game fine tunes challenges that he can meet. The elegance of this method is that the user encounters different scenarios specific to his condition every time he restarts the game. For example, if the game is too hard, the player gets more weapons, recovers life points faster, or faces fewer opponents

While employing the DDA mechanism, certain factors need to be taken into consideration like the difficulty should be adjusted moderately to avoid the rubber band effect. Rubber band effect can be observed in the racing games if dynamic difficulty balancing is not moderate. In that the AI's vehicle will become significantly faster when behind and would be significantly slower when in lead. DDA has been employed in modern games frequently nowadays. The latest game to employ the technique is Resident Evil 5[4] where they deployed a difficulty scale which graded the user's performance on a number scale 1-10 and adjusts the both the enemy attacks/behavior used and enemy damage/resistance based on the players' performance. The Step 2 of the above algorithm can be implemented in various ways and popular approaches include the way we stated which is also called Hunicke and Chapman's approach [3], the sue of behavior rules, reinforcement learning(RL), use of genetic algorithms, artificial neural networks (ANN) and fuzzy neural networks. Considering the above, DDA still has some major flaws-

1. Players can learn to exploit DDA by pretending to be worse than they are.
2. DDA doesn't work for all kinds of challenges. For example let us take a sports game, by applying this mechanism the difficulty level will keep on swinging from amateur to world class. This would be actually like betraying the user to an extent by letting him win for the sakes of it. A sports game has to be set to a pre-defined level and one cannot change it mid game. A world class team cannot play poorly just to make the game even.
3. DDA can create absurdities. E.g. Rubber Band Effect.
4. DDA ruins pacing and obviates good level design.
5. DDA is not that subtle. The user often figures out that everything is being done according to his behavior.

4. DYNAMIC ADAPTABILITY USING DATA ANALYSIS (DADA)

To rectify the aforementioned drawbacks a new approach is proposed. This approach focuses on the main goal which is dynamically adapting to the user while keeping the pre-set

difficulty levels same. While each game is different to one another, the user whether experienced or novice finds a certain way to play and also almost every time approaches a particular game in a set way. This leads to the emergence of a pattern in the user behavior. This pattern holds the key for dynamic adaptation of an AI. The studying of this pattern will help the AI act as if it is a human entity which is capable of analysis and rational thinking. This analysis will provide the AI countermeasures which will result in the fulfillment of our goal. Now, the proposed approach starts with deciding a time interval after which a pattern will be stored. This pattern constitutes every move that the user makes in the stipulated time interval. Along with this a critical or a key aspect of the game needs to be identified which will constitute a sub-pattern that needs to be analyzed immediately irrespective of the time interval decided. These critical events could be anything from scoring of a goal in a football game to defending an AI attack in a RTS. These patterns obtained are then associated with an existing pattern for reducing the overhead of processing the same results again. If no such pattern exists with us or if the pattern association fails (i.e the pattern in question is new) then it is stored. This can be achieved in varied ways but popular approaches include the use of Neural Networks models. This step is followed with the 'Feature Extraction' phase which will extract features or attributes from the pattern. The feature extraction mirrors with the heuristic function (h(n)) which will help us reach the goal faster. After this phase is complete, analysis must take place on the features that have been extracted. This analysis will help us understand the user play and will help give the game AI the added feature of 'rational thinking'. Now once the analysis is formulated, the AI must take action on it or produce a solution to help counter the set user behavior. This pattern and its counter measure together are stored in database for future reference. This entire process can be characterized as dynamic learning, as the game learns about the user behavior and devises its own plan to help counter it.

Let us illustrate the algorithm with the help of a football game FIFA. The steps of the proposed algorithm proceeds as follow:

A. Step 1:

The algorithm starts by deciding a time interval and storing a pattern. Define time interval 'T' after which you want to analyze a pattern. This time interval can be decided with respect to the pre-set difficulty level. The harder the difficulty level, shorter the time interval. This time interval that is decided constitutes a main pattern, but since this is a football game the critical aspect is goal scoring and then that pattern constitutes a sub pattern which needs to be acted upon immediately. The main pattern is analyzed after time interval 'T' whereas the sub pattern is analyzed as soon as it is stored. So the total number of patterns obtained after time 'T' are:

$$\sum p(\text{total}) = p(T) + p(g)$$

Where p (total) = total pattern,

P (T) =pattern after time t.

P (g) =Pattern for each goal.

If value of p(g) =0 then no goal is scored in time 'T' else p(g)=number of goals scored.

Both the patterns follow the below steps:

B. Step 2:

The pattern consists of huge amount of data and the need is only for the important aspects from it to make the analysis easier. Hence the extraction of few features (or attributes) from that pattern could be:

a) Main pattern for time 'T':

- | | |
|---------------|-------------|
| 1: Possession | 2: Passing |
| 3: Shooting | 4: Crossing |
| 5: Dribbles | 6: Defense |

b) Goal Pattern:

- | | |
|-----------------|--------------------------|
| 1: Side | 2: Pass to goal |
| 3: Way of goal | 4: Goal scoring position |
| 5: Type of goal | |

C. Step 3:

Now in the next step, analysis of the game play of the user is undertaken so that the AI combats the player by changing the AI and thus making it adapt to the user's game play by not changing the difficulty level. The features extracted in the previous step provide a guideline to the overall solution. For example, the aforementioned feature of 'Total number of passes' helps the AI device an overall scenario of that user. The high number of total passes indicates that the user is proficient in the passing or prefers to play a passing game. Similarly the passing accuracy feature substantiates the user's proficiency in passing. All the above features are similarly analyzed with each providing a certain conclusion about the game play of the user.

D. Step 4:

The conclusions that are formed in the previous step need to be acted upon to enhance the user's game play experience. The counter measures to these could be the use of the in game techniques that have already been implemented for regular game play. For example a user whose characteristic game play is based on short passing could be provided with the counter measure of high pressing. This counter measure as mentioned above already forms a part of regular game play and is already implemented as a part of overall aspect of the game. This step just reuses that aspect in the current situation which otherwise would not have happened.

E. Step 5:

The pattern that is obtained and the counter measure selected in the above step together are stored in the database for future pattern association.

To illustrate the above technique we take following scenarios:

- a) Consider a player that scores a goal with header by receiving a cross from right side. The features extracted for this sub pattern are:
1. Side from which goal was scored
 2. Goal Scoring position
 3. Way in which goal scored
 4. Type of goal

Upon analysis of these features the following answers to the extracted features are obtained (which constitute the phase of data analysis)-

1. Side from which goal was scored : Right

2. Goal Scoring position: 6 yard
3. Way in which goal scored: Header
4. Type of goal: Open play

From the above we can deduce that the user relies on right side (wing) more rather than other areas. The countermeasure to this can be to double up players on the right side to stop the long pass for the goal. Now this countermeasure is already implemented in the game maybe as a part of higher difficulty level so applying them is a trivial issue.

1. Total Passes: 100
2. Passes Completed: 80
3. Short Passes: 70
4. Long Passes: 10

From the above extracted features the AI is instructed to adapt to this problem by pressing more high up the pitch. Also the AI is told to make more tackles, interceptions and marking the players which might receive the ball. Again as stated before the countermeasure is a part of the game itself.

- b) The above was a critical scenario where goal was scored. The other scenario would be for stipulated time. The scenario is that the user relies on short passing more to build up its game. The features extracted are:
1. Total Passes
 2. Passes Completed
 3. Short Passes
 4. Long Passes

The above algorithm can be employed to the games of other genres as well. Only the features would vary and time 'T' can be adjusted, the remaining algorithm and steps remain the same. For example, for strategy games the features that should be extracted might be the resources gathered, units created, research done, buildings constructed etc. There are certain disadvantages to the method we have proposed such as there is storage overhead also coupled with memory and requires high processing speed.

Say, the analysis provides the following details-

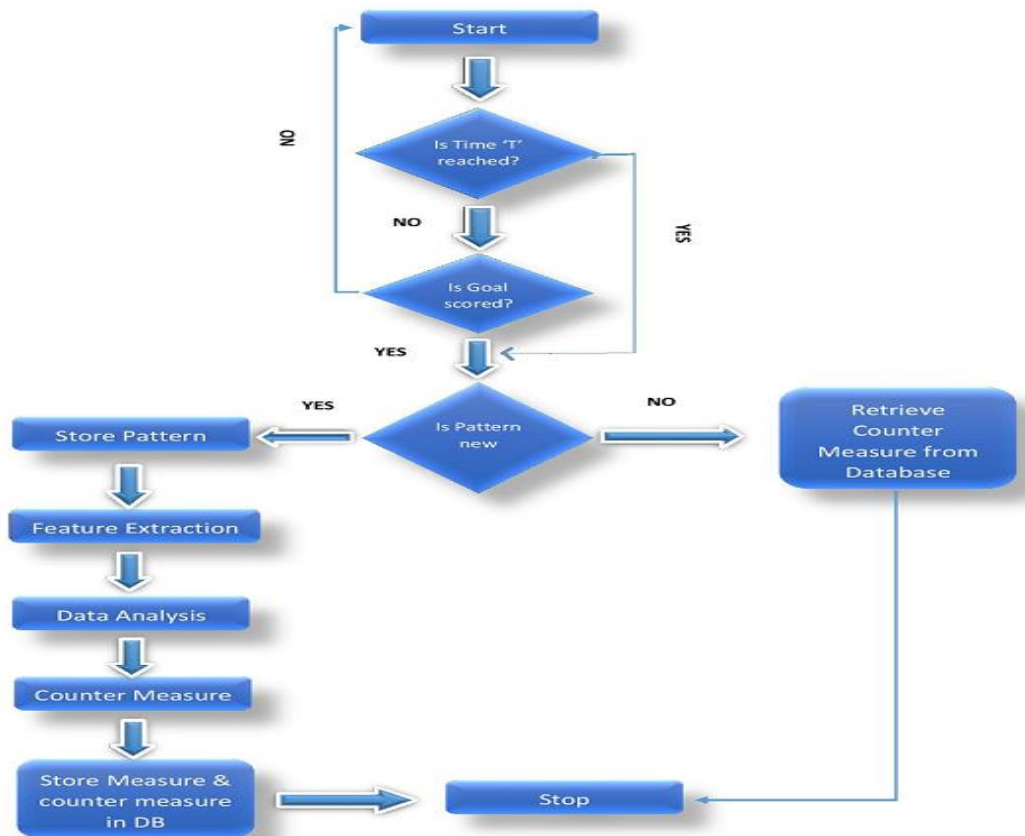


Figure 4: DADA Algorithm

5. DADA ANALYSIS

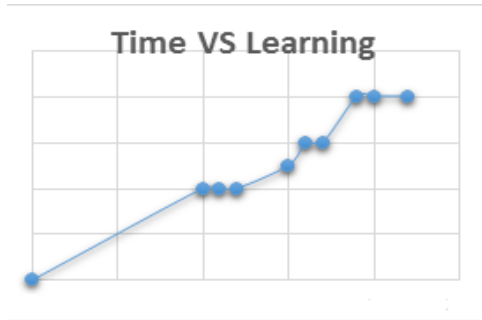


Figure 2: Time Vs Learning

Figure 2 shows that the learning line keeps on increasing because every time a user feels that he has combated an AI, the AI changes its game play and thus force the user to keep on learning to combat the AI.

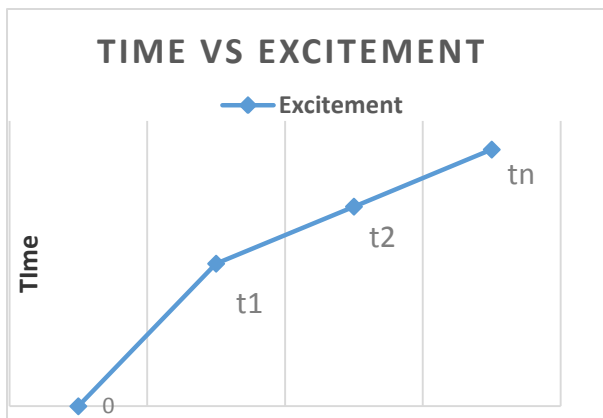


Figure 3. Time vs Excitement Graph

The Figures 3 depict the changes that occur after employing the proposed algorithm in which excitement level keeps on increasing as new challenges are tackled by the user

6. CONCLUSION

This phase of game development is calling for new and improved techniques to enhance the AI to complement the already sophisticated graphics so as to raise the user experience to an all new level. The 'bot' or the computer player does not possess the ability to analyze the situation and react and redefine their strategies accordingly. In this paper two techniques were discussed to do the same. DDA provides a simple and elegant solution to adapt the AI to the user, constantly changing the AI to cater new and exciting user behavior.

The application of DDA is extremely difficult as many genres of games fail to provide the similar level of details and the maneuvering of the game details, bots and environment is not feasible in many as in FPS. These shortcomings highlight the need for another technique that caters to a variety of genres while following a traditional and classical AI approach. The

proposed method, DADA attempts to learn user behavior figuring out what he is good at, what he finds difficult and what he fails at. This can be achieved by analyzing the 'user pattern' extracting valuable and critical features from them. This scalability of the method is what makes it appealing. The application of the method to a wide variety of genres of the games along with the extensive yet feasible data analysis and providing an in-game solution makes this approach highly recommendable.

The need of the hour is a dynamic AI which transcends the boundaries that exist between a human and a bot. While an average user is moderately satisfied with his experience, many of the faults of the AI including the repetitive behavior are spotted far too easily by him. The assumption that there are going to occur such faults as the entity in question is a 'bot', highlight the issues with the AI which was developed to precisely eliminate this notion. The proposed method has been designed keeping in mind the same and we hope it provides an easy yet elegant solution to the aforementioned issue.

7. REFERENCES

- [1] Schwab (2004). AI Game Engine Programming. Charles River Media. ISBN 1-58450-344-0
- [2] EA Sports, FIFA™, <http://www.easports.com/uk/fifa>.
- [3] Robin Hunicke, V. Chapman (2004), "AI for Dynamic Difficulty Adjustment in Games". Challenges in Game Artificial Intelligence AAAI Workshop. San Jose. pp. 91–96
- [4] Resident Evil 5 Official Strategy Guide. Prima Publishing. 5 March 2009.
- [5] Fabio Aiolli and Claudio Enrico Palazzi, "Enhancing Artificial Intelligence in Games by Learning the Opponent's Playing Style"
- [6] Buckland (2002). AI Techniques for Game Programming. Muska & Lipman. ISBN 1-931841-08-X.K. Elissa, "Title of paper if known," unpublished.
- [7] P. Spronck, I. Sprinkhuizen-Kuyper, E. Postma (2004), "Difficulty Scaling of Game AI". Proceedings of the 5th International Conference on Intelligent Games and Simulation. Belgium. pp. 33–37.Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," *IEEE Transl. J. Magn. Japan*, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
- [8] Koutroumbas, Konstantinos; Theodoridis, Sergios (2008). Pattern Recognition (4th ed.). Boston: Academic Press. ISBN 1-59749-272-8.
- [9] Lewis-Beck, Michael S. (1995). Data Analysis: an Introduction, Sage Publications Inc, ISBN 0-8039-5772