Dynamic Difficulty Adjustment in Video Games

⋆

Diana Sofía Lora Ariza

Dep. Ingeniería del Software e Inteligencia Artificial
Universidad Complutense de Madrid, Spain
dlora@ucm.es

Abstract. Dynamic difficulty adjustment (DDA) in video games context allows a game to be automatically adjusted and customized based on the player's performance. Our goal is to build a DDA architecture able to learn from behavioral data and return appropriate feedback to maximize the player's performance and enjoyment. For our experiments, we use Tetris Analytics video game, which is like a traditional Tetris from the player's point of view. However, it allows us to extract data from the games such as the movements made by the player, the state of the board, next piece, etc. With this data, we predict the player's profile and personalize the game.

Keywords: Dynamic Difficulty Adjustment · Time Series · Video games · Tetris · Case-based Reasoning · K-Nearest Neighbor.

1 Problem

A majority of the game content is static and created during the development phase of video game production. Game designers attempt to make game content that is challenging but not overly difficult for the average player in the target audience. For this reason, feedback from players is essential to help game designers to distinguish from an exceedingly difficult task to a great challenge [6]. Computer games need to provide appropriate responses to changing circumstances and perform complex behaviors in realistic environments [1]. Mainly to keep players engaged, happy, and playing the game more and buying more games by the company. Games require a certain level of "intelligence" to remove the need to expensive and time-consuming manual editing of game difficulty by game designers. It also helps players that fall outside of the "average" to still have fun. And on top of that, it ensures a more "life-long" consistency to the difficulty so that the game will keep changing over-time to maintain the perfect challenge.

⋆ Supported by the Spanish Ministry of Science and Education (TIN2017-87330-R)

2 Objectives

2.1 General Objectives

Dynamic Difficulty Adjustment is a new subject compared to other application areas in the field of video games. It can be used to automatically alter the game's difficulty to meet the player's expectations [2]. It is complex to know the skill level of a player with one decision. Therefore we use time series, where we can analyze the evolution of the player in a time frame. We also employ CBR because it works with little data. It is a simple technique to explain since it works with experience. And in case there is a lack of knowledge in the experience data, then the addition of expert knowledge domain complements it.

Our research focuses on using *Theory of Flow, DDA, CBR and time series in video game context to create a DDA architecture* that improves the experience of the player by adapting the game to the level where it is challenging but not overwhelming.

2.2 Specific Objectives

- Review DDA state of the art in video game context.
- Evaluate novel techniques for the categorization of behavioral data.
- Use time series to identify the evolution of a player in a specific time range.
- Use categorization techniques for player profile prediction.
- Review The Theory of Flow and its application in video game context.
- Implementation of DDA in real game.
- Execute experiments to analyze DDA efect in players' experience.
- Create DDA architecture that allows abstraction of the fundamental processes for game adaptation.

3 Research Plan

The planning of our research divides into three stages. The first stage focuses on the review of the state of the art and application of knowledge acquired in a simple video game called *Tetris Analytics*. This stage takes place in the first two years, and it is complete. The highlighted activities of this phase are the following:

- Review of the state of the art of machine learning techniques.
- Review of techniques applied in video games for players categorization and prediction of their skill level.
- Perform experiments of player classification and dynamic difficulty adjustment based on user interaction in *TetrisAnalytics* using the techniques reviewed previously.
- Compare performance of evaluated techniques.
- Publication of findings.

The second stage focuses on the establishment of a methodology for players categorization and dynamic difficulty adjustment based on user behavior in video games. This stage is partially completed and must finish next year. The main activities are:

- Establish general processes used in *Tetris Analytics* that can use in other games.
- Establishment of a methodology for a player's categorization and dynamic difficulty adjustment in video games.
- Publication of findings.

The third stage focuses on the application of our findings in a more sophisticated video game. This stage should finish within two years, and we have not started yet. The main activities of this stage are:

- Conducting experiments with a more complex game for players categorization and dynamic difficulty adjustment based on user interaction.
- Verification of methodology.
- Publication of findings.
- Writing the thesis.
- Preparation and presentation of the thesis.

4 Current Research State

4.1 Tetris Analytics

We used *Tetris Analytics*, a version of Tetris implemented in Java by a member of our research group. Our Tetris game looks like the original one for the player point of view but internally provides extra functionality to extract, store, and reproduce game traces. Tetris is a widely known game, where the difficulty is always rising, so it is easier to evaluate the stress level of a player, and it is a simple game to adjust. Therefore we can extract behavioral data, select the most significant features to characterize the playing style of each player, determine their skill level, and dynamically adjust the difficulty of the game.

Each time a new piece appears on the top of the board, players have to make two different decisions. The first one, the *tactical* decision is to select where to settle the piece. The second decision involves all the *moves* (translations and rotations) required to lead the piece to that final location. We only consider the tactical decisions to define the skill level of the player, but we are aware that we could also extract valuable information from the concrete moves.

4.2 Player Profile and Cases

We used clustering to analyze data extracted from TetrisAnalytics. My work on this resulted in a publication at The Florida Artificial Intelligence Research Society [5]. From these experiments, we were able to determine 3 player profiles and their score ranges:

- *Newbie* players score between 0 to 2999
- *Average* players score between 3000 to 5499
- *Expert* players score 5500 or more

Then, we notice that it was difficult to predict players' profiles during gameplay by analyzing individual tactical decision. It was better to analyze the evolution of the player in a time range. So, we started to group tactical decision and use CBR to find the most similar case in the case base of past games [3,4]. We tried different ways to create our case base and found that the most effective approach is to take ten consecutive tactical decision. Each case contains the following data:

- Time or number of the piece falling through the board.
- Time series with the evolution of the score.
- Time series with the evolution of the total empty space on the board.
- Time series with the evolution of the maximum height reached by any piece on the board.
- Player profile or skill level.
- Flow level: a score describing how good the game was for the player.

Note that the game computes the last two features once it finishes. After a game ends, we asked the player on how was their experience during the game. The flow level was a score given by the player on a Likert scale of 5 values.

To calculate player profile during gameplay, we use a query case Q describing the evolution of the variables during the last ten tactical decision and use a K-Nearest Neighbor classifier (k-NN) algorithm to get similar cases from our general case base. The similarity between the two cases is the linear combination of the similarities between their time series. We use a simple similarity measure based on the Euclidean distance to compare time series. The similarity function only considers cases that have the same time range in previous games. Each time series has a weight or importance. The most significant time series is the score (70%), followed by the number of holes in the game board (25%) and finally the max pieces' height (5%).

A majority vote decides the current player's profile among the k most similar cases. To avoid extreme profile changes, we use an inertia function that eases the transition from newbie to expert and vice versa. Every time the game predicts the player's profile, the inertia function is applied, and it compares the current player profile with the new one.

4.3 Best Next Piece

To select an excellent piece for the player, we compute all possible ways to place each type of piece and use a heuristic to decide how good those likely future boards are. The heuristic tries to maximize the score (i.e., complete newlines) and minimize the number of holes and height of the board. Then we assign to each piece the rating of the best game board that can be reached placing the piece optimally. Finally, we randomly choose one of the three best options and use it as the next game piece. Now, we only worry about making the game more relaxed, but it is in our plan also to make it harder for more experienced players.

4.4 DDA System Architecture

Now, we are working on the second stage of the research plan, trying to extract the main processes from dynamic difficulty adjustment in Tetris Analytics and create a *DDA system architecture*. Ideally, from this architecture, we will create an abstraction of the processes involved in dynamically adjusting a game.

Each user has a different way of playing, performance level and thoughts in what would be a game that he would enjoy to the fullest. The idea is to create a *stress level* that maximize the user experience during gameplay. For this purpose, we create a *user experience flow* that takes advantage of dynamic difficulty adjustment benefits. In Tetris Analytics we change the game by giving the best next piece that helps the player make lines [4], we use a case base that help us determine if the player needs help and which would be the best next piece according to their performance.

References

1. Charles, D., Kerr, A., McNeill, M., McAlister, M., Black, M., Kcklich, J., Moore, A., Stringer, K.: Player-centred game design: Player modelling and adaptive digital games. In: Proceedings of the digital games research conference. vol. 285, p. 00100 (2005)
2. Jennings-Teats, M., Smith, G., Wardrip-Fruin, N.: Polymorph: dynamic difficulty adjustment through level generation. In: Proceedings of the 2010 Workshop on Procedural Content Generation in Games. p. 11. ACM (2010)
3. Lora, D., Sánchez-Ruiz, A.A., González-Calero, P.A.: Difficulty adjustment in tetris with time series (2016)
4. Lora, D., Sánchez-Ruiz, A.A., González-Calero, P.A.: Time series and case-based reasoning for an intelligent tetris game. In: International Conference on Case-Based Reasoning. pp. 185–199. Springer (2017)
5. Lora, D., Sánchez-Ruiz, A.A., González-Calero, P.A., Gómez-Martín, M.A.: Dynamic difficulty adjustment in tetris (2016)
6. Pagulayan, R.J., Keeker, K., Wixon, D., Romero, R.L., Fuller, T.: User-centered design in games. CRC Press Boca Raton, FL (2002)