Measuring Control to Dynamically Induce Flow in Tetris

Diana Sofía Lora-Ariza, Antonio A. Sánchez-Ruiz, Pedro Antonio González-Calero, and Irene Camps-Ortueta

***Abstract*—Dynamic Difficulty Adjustment (DDA) is a set of techniques that aim to automatically adapt the difficulty of a video game based on the player's performance. This paper presents a methodology for DDA using ideas from the theory of flow and case-based reasoning (CBR). In essence we are looking to generate game sessions with a similar difficulty evolution to previous game sessions that have produced flow in players with a similar skill level. We propose a CBR approach to dynamically assess the player's skill level and adapt the difficulty of the game based on the relative complexity of the last game states.**

We develop a DDA system for Tetris using this methodology and show, in a experiment with 40 participants, that the DDA version has a measurable impact on the perceived flow using validated questionnaires.

***Index Terms*—Dynamic Difficulty Adjustment, Flow, Artificial Intelligence, Video Games.**

I. Introduction

THE video game industry has grown rapidly in recent years. In 2020, this industry generated revenues above $170 billion and it is expected to continue growing in the following years. This growth has motivated investors to finance with additional resources the development of new titles, and game studios to look for new ways to overcome the inherent difficulties involved in creating and maintaining good quality video games. The production of an average AAA video game takes between 1-3 years not just because the time required to create all the assets and game mechanics but because all these ideas must be tested continually to validate that they really work and the game is entertaining. This way, game designers play a fundamental role in the production of video games. They are usually in charge of defining the game levels, the puzzles, the controls, the game mechanics, the character dialogues, ... and especially the difficulty levels of the game that usually involve defining different sets of behaviors for the AI characters that will be used according to the player's skill level [1]. In other words, the "intelligence" seen in most video games today is the result of trying to anticipate different player behaviors and then, during the production stage, to implement a standard set of actions to respond appropriately in each expected scenario. Most of the content in video games today is predefined, created during the development process, with the hope that it will be adequate for most of the players, challenging but not too difficult, but that is not always achieved. In fact, player's feedback is an essential tool for designers to distinguish between extremely difficult tasks and a great challenge [2].

An "intelligent" video game should be able to decide what to do in scenarios that the designers could not anticipate and provide an appropriate response [3]. In particular, the classical predefined difficulty levels present in most video games are not optimal for all players, and the difficulty of the game may not evolve at the same rate as the player's skills, making the game too easy or too difficult. In this sense, *difficulty* should not be understood as a static property, but a subjective factor derived from the interaction between the player and the proposed challenge [4]. Dynamic Difficulty Adjustment (DDA) is a set of techniques that aim to automatically adapt the difficulty of the game based on the player's performance [5]. That is, the player-game interactions are continuously evaluated to change the difficulty of the game according to the player's current needs with the goal of keeping the player engaged. A notable advantage of using DDA is the reduction in video game production costs, because if the game can adapt itself, then designers and developers require less effort trying to anticipate all possible situations [6]. For this reason, DDA has captured the attention of leading companies in the video game industry. Lopes and Bidarra [7] surveyed the state of art of adaptivity in games and simulations, from both academia and industry, and concluded that adaptivity is establishing itself as a rapidly maturing field and that current advances show good results in adapting towards an optimal challenge level and emotional states such as fun, frustration, predictability, anxiety or boredom.

Flow is a mental state in which a person performing some activity is fully immersed in a feeling of focus, full involvement, and enjoyment, usually resulting in a transformation in the sense of time. The connection between flow and immersion in video games is quite direct since the ability to transport its players to their personal flow zones has been identified as a requirement for a well-designed game [8].

This paper presents a novel methodology for DDA using ideas from the theory of flow [9] and case-based reasoning (CBR) [10]. A key advantage of using flow theory is that, since it is a very popular theory in Psychology research, there are different instruments to measure it, both on the basis of physiological measurements and posteriori questionnaires. Those instruments can be used to identify previous games that have produced flow in players. Our methodology aims to generate flow by altering the difficulty of the game so that its evolution is similar to previous games that have produced flow in players with a similar skill level. We use a CBR approach to dynamically assess the player's skill level and adapt the

The authors are with the Universidad Complutense de Madrid, Madrid 28040, and Universidad Francisco de Vitoria, Madrid 28223, Spain (email: dlora@ucm.es, antsanch@ucm.es, pagoncal@ucm.es, irene.camps@ufv.es).

Manuscript received July 31, 2021; revised January 25, 2022.

difficulty of the game using a measure of control based on the relative complexity of the current game and a set of previous similar labeled games. We have implemented a DDA system for Tetris and show, using validated flow questionnaires, that has a measurable impact on the players.

The rest of the paper runs as follows. In next section we briefly introduce the theory of flow and presents its relation to DDA in games. Then, in Section III, we present the case-based DDA methodology exemplified with Tetris. Next, we describe an experiment that shows that our approach effectively improves the sense of flow in players in Section IV. The paper concludes with some conclusions and future lines of research.

II. Related Work

A. The flow theory

Czikszentmihalyi [9] defines flow as *the optimal experience when nothing else matters*. Entering flow depends on creating the perfect balance between the perceived challenges of the task at hand and one's perceived abilities. Nakamura et al. [11] say that it is the subjective challenges and subjective skills, not the objective ones, that influence the quality of a person's experience. In other words, the person must have confidence in their capacity to complete the task successfully to enter flow. Csikszentmihalyi establishes nine dimensions which together represent the optimal psychological state of flow [12]: challenge-skill balance, merging action-awareness, clear goals, unambiguous feedback, concentration on the task at hand, sense of control, loss of self-consciousness, the transformation of time and autotelic experience.

Nakamura et al. [11] state that attention plays an essential role in achieving a lasting feeling of flow because a person's attention plays a part in the type of emotions that the task can generate. A person experiences apathy or boredom when the task at hand is simple, as their attention is away from the task. On the contrary, a person feels anxiety when they have their full attention on the task, but it is very complex. That is, the challenge exceeds their capacities. In an ideal situation, the player has full attention on the task, its difficulty level is correct to the player's capacities, and the player has a feeling of control, then they can enter flow. As a person masters a challenge, her skills increases and thus the challenge must grow in difficulty along with the person to keep them in flow or, otherwise, the task ceases to be enjoyable.

Figure 1 shows in more detail the eight emotions a person may experience while performing a task. The state of flow occurs when the task at hand is exciting and challenging but achievable, between the emotions of control and arousal. These feelings intensify when the challenges and skills required to perform the task are beyond the player's average levels.

Descriptions of the flow experience in general tasks are identical to those experienced by players when immersed in games, such as losing track of time [8]. Cruz et al. [13] review previous attempts to adapt the flow theory in the context of video games. They describe different mappings between flow elements and game features in order to promote player-centered game design, and foster flow in the player experience.

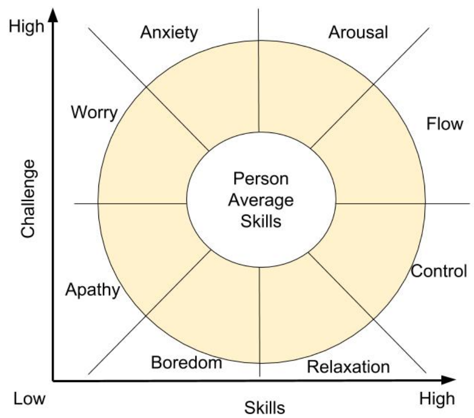

Fig. 1. Mental state in terms of challenge level and skill level, according to Csikszentmihalyi's flow model [11].

Of particular relevance in establishing the connection between flow theory and game design is the GameFlow model [14]. GameFlow is a model of player enjoyment, comprised of a set of criteria derived from games user experience literature and structured into eight elements that can be mapped to Csikszentmihalyi's concept of flow.

B. Flow and DDA

Most attempts to connect the flow theory and DDA make use of devices that monitor physiological variables (electroencephalograms (EEG), heart rate or galvanic skin response among others) to identify the player's emotional state and adjust the difficulty of the game accordingly. Chanel et al. [15] describe an early attempt on identifying emotions when playing Tetris at three different difficulty levels. Klasen et al. [16] assess how the brain responds to the different dimensions that contribute to the flow experience in a first-person shooter game, showing that the balance between challenge and skill, sense of control and concentration are most representative of the flow experience. One of the first applications of emotion detection to actually implement a DDA system is described in [17] using players' EEG signals during playing a rhythm game. More elaborated versions, both for shooter games, are described in [18] and [19] where the authors use different devices to monitor physiological signals and adapt the difficulty of the game to maintain the player's level of excitement and improve the gaming experience.

Although recent projects like BITalino have lowered the entry barrier of bio-signal acquisition, the task of obtaining bio-signals of good quality is time-consuming, and typically tools for obtaining such measures are not comfortable for the user, requiring electrodes attached to different parts of the body. For that reason, it is not foreseeable that gamers will become accustomed to using bio-signal acquisition hardware on a regular basis in the near future. As an alternative to physiological variables, numerous self-report questionnaires can be found in the literature of Psychology that have been validated as instruments for assessing flow [20]. In this work, we propose the use of ex-post questionnaires instead of real-time difficulty adjustment through emotion detection using physiological measures.

C. Other approaches to DDA

Most of the related work in DDA defines specific techniques for adapting difficulty in different types of games. For example, [21] tested five different algorithms for adjusting difficulty in Tetris that, instead of adjusting game speed, adjust difficulty by choosing blocks based on the current game state. Adapting the AI of non-player characters (NPC) is the focus of several contributions, such as [22] that propose an algorithm for adjusting spawn rate and distance of the enemies, [23] in MOBA games, or [24] and [21] in racing games. Sophisticated techniques such as reinforcement learning and evolutionary computation are typically applied in the generation of NPC that can dynamically adapt to the player. One distinguishing feature of the work presented in this paper is the focus on ”when to adapt” and not so much on ”how to adapt” the difficulty.

In [25] the authors present an initial effort in developing a universal mechanism for DDA, an online learning algorithm that takes as input game-specific ways to modify difficulty and the current player's in-game history and produces as an output an appropriate difficulty modification. Nevertheless, the paper just covers the initial step of clustering into different types and predicting the type from short traces of gameplay. Also for player modeling, [26] presents a game-independent method for predicting player performance. They proposed a data-drive tensor factorization approach that can predict changes in players' skill mastery over time. Nevertheless, they only cover the initial stage of player characterization. In general, the difficulty adjustment phase is mostly game-specific and may be more or less laborious depending on the game. In this work, we decide when to adjust the difficulty based on the player's skill level and evolution of the complexity of the current game compared to other previous good and bad games, which also connects our work with previous work on player modeling.

III. Case-based DDA

In this section we introduce a general methodology that serves as a guide for implementing DDA during the video game development process and explain how we have implemented its different phases in Tetris. Our goal is to facilitate the creation of a system to improve the player's experience by balancing her real skills and the skills required to complete the proposed challenges.

Our methodology consists of 2 phases (see Figure 2). The first one, *players' pre-analysis*, is an “offline” phase in which we collect a set of games played by different players and then we analyze those games to learn about how different players interact with the game and which game variables are useful to describe the player's performance during the game. This is an exploratory stage that will help us to create and annotate the data used by the DDA system. The second phase is “online” in the sense that it is performed during the game, and consists of 2 main components, *players' categorization* and *difficulty adjustment*, that are executed one after the other continuously during the game. The *player categorization* component will predict the skill level of the current player *during the game* comparing her performance with the performance of other previous players. The *difficulty adjustment* component, in turn, will decide if and how to adapt the difficulty of the game to meet the current player's skill level.

Note that these phases are very general and can be implemented using different techniques. Moreover, the features used in each game to model the player's performance and to alter the difficulty will be specific for each game. For this reason, game designers play an important role in our methodology and should be involved to identify the right game variables to model the player's performance and dynamically adapt the difficulty of the game.

In the following sections we explain in detail how we have implemented this methodology in Tetris. We use a case-based reasoning approach (CBR) [10] in which cases represent the evolution of previous games during short time windows, and we focus on 2 flow dimensions that have been identified as especially relevant to induce flow in video games: sense of control and challenge-skill.

A. Players' pre-analysis

The goal of this phase is to collect a set of game traces and analyze them using different statistical and visualization techniques to understand how different players interact with the game, differentiate games that are more and less satisfactory for the players, and identify the game variables that better assess the performance of the player during the game.

1) Collect game dataset: The goal in this step is to collect enough game traces and evaluations of those games to represent the potential population of players. In asynchronous setups where players participate at different times without the supervision of some moderator, it is important to choose a very quick way to evaluate players' satisfaction after each game because, while people were willing to play a few games of Tetris, they were less willing to fill out a survey after each game. Anything that takes more than a few seconds may discourage the participants from playing another game. On the other hand, synchronous setups are more expensive and usually involve less participants, but the presence of a moderator allows to use more sophisticated satisfaction surveys and collect more data about each player and game.

We used an asynchronous setup and reached potential participants sending emails to different mailing lists in our university. The target participants were grad students in their 20-30s from different Faculties (sciences, engineering, literature, ...) and with different experience playing video games. We asked them to assist us with a research study about video games in which they only had to play Tetris and answer one simple question. They were not incentivized to participate in any other way. They played a special version of Tetris called Tetris Analytics that looks like a normal Tetris game but stores the game traces in files so that they can be reproduced and analyzed later using different visualization tools and machine learning algorithms. The participants could play as many

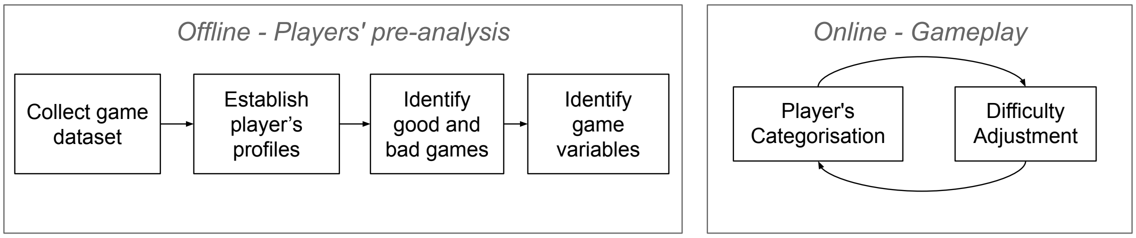

Fig. 2. DDA methodology scheme. The goal of the offline phase is to collect, analyze and annotate a set of games that will be used in the second phase. The online phase consists of 2 components that continuously analyze the player performance during the game and adapt the difficulty accordingly.

games as they wanted. The new pieces that appeared in the game were randomly selected and there was only one standard difficulty level. After each game, the system asked the players to assess their experience evaluating the sentence *It was a good Tetris game* with a 5-value Likert scale, where 1 means strongly disagrees and 5 strongly agrees. We collected 156 game traces played by 51 different players. 9 games were evaluated with a score of 1, 29 games with 2, 28 games with 3, 55 games with 4 and 35 games with a score of 5.

2) Establish player's profiles: In several games, and in Tetris in particular, there is a final score that measures the performance of the player during the game. We can use this variable to label player's profile or skill level but only when the game has ended. Also, it is not obvious what score ranges correspond to each profile or how many profiles should be differentiated. Fortunately, we can use the data collected in the previous step to make reasonable choices. We used k-means to cluster the final scores and the elbow method to select 3 clusters that represent different profiles that we identify with *beginners* (0-2999 points, 72 games), *average players* (3000-5999 points, 82 games) and *experts* (more than 6000 points, 2 games). Then, each game was annotated with a profile.

Note that we can create more or less profiles depending on the desired granularity. We can use a number of profiles that makes sense for the game designers or use some algorithmic approach to select an optimal number of clusters.

3) Identify good and bad games: During the collection of the game dataset, each game was evaluated with a satisfaction score using a 5-value Likert scale. We consider that a game was *good* if it was evaluated with a score of 4 or 5, and it was a *bad* if it was evaluated with a score of 1 or 2. A score of 3 means that the player did not feel the game was especially good or bad, so we do not consider them in this classification. Additionally, bad games are divided into two subcategories, *too hard* and *too easy*, depending on the final score of the game with respect to the scores of the players within the same profile. For example, if an average player evaluates a game with a bad satisfaction score and the final game score is much lower than the mean score of average players, we assume the game was too difficult and that produced a negative effect in the player's experience. 90 games from the dataset were annotated as good games, 19 as too hard and 4 as too easy. We will use these 3 subsets of games later to adjust the difficulty of the game so that it becomes similar to the good ones and different to the bad ones.

4) Identify relevant game variables: The purpose of this step is to select some variables that represent the player's performance during the game, and to identify the relevant moments of the game in which we should measure them. We will represent the evolution of these variables by means of time series describing the evolution of the game. We will then divide these time series into *performance windows* that represent the evolution of the game over short periods of time. This way, we will be able to adapt the difficulty in new games comparing the last performance window of the current player with the annotated performance windows in our dataset.

In Tetris we distinguish two types of decisions. When a piece appears at the top of the board, the player must analyze the current board and decide where she wants the piece to end up. We call these decisions tactical decisions, since they lead to states that the player wants to reach in the near future. Then, to reach those states, the player must decide all the intermediate actions (translations and rotations) needed to bring the current piece from the top of the board to the desired final position. In this paper we will focus on tactical decisions, since we think they represent the most significant moments of the game, and we will measure the evolution of the game only when each piece settles in its final position. We leave for future work to take into account also the intermediate moves that lead to the final position, which undoubtedly contain relevant information, but are also lower-level and noisier (e.g. nervous players change the rotation of the piece back and forth several times).

The selection of the variables that will be used to predict the current player profile *during* the game is a trial-and-error process that must be done with the help of the game designers. Based on our knowledge of Tetris we have selected the following variables to represent the evolution of the game:

- *Piece number* from the start of the game. As we only consider tactical decisions and there is one tactical decision per piece, it represents time.
- *Score* accumulated by the player once the current piece is settled.
- *Board height* or the highest row occupied by a piece.
- *Holes* or empty cells that are completely covered by another full cell.

Figure 3 shows how these game variables evolve in a game of an average player. This player was able to settle 64 pieces before the game finished and got a total score of 2480. The yellow line describes the score evolution that grows slightly

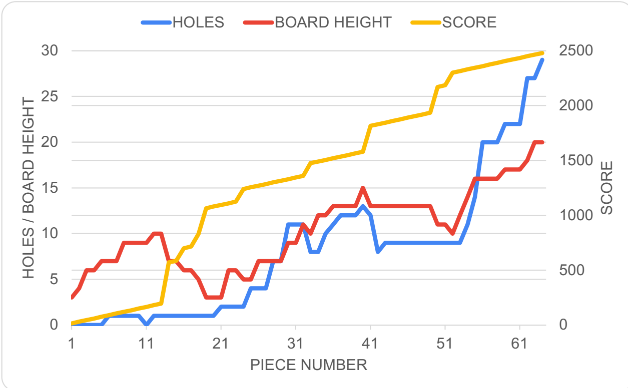

Fig. 3. Evolution of the game variables for a beginner player.

with each new piece and more abruptly when the player makes a new line. The red line represents the board height and when the game ends when it reaches 20. We can see that during the first 30 pieces the player played in the lower half of the board, but then the board height grew up because the pieces fell too quickly for the player to set them compactly. Similarly, the blue line represents the number of holes.

The evolution of each game variable can be represented as a time series in which each value corresponds to the value of the variable in consecutive instants of time. Since we are interested in measuring the player performance in different moments of the game, we divide the time series in *performance windows* that represent the evolution of all the variables during a short period of time. This way, each performance window is a matrix of $m \times n$ where m is the number of variables and n is the length of time window.

B. Player's Categorization

We use a case-based reasoning (CBR) approach to decide the skill profile of a new player during the game. The description of each case contains a performance window of length $n = 10$ (time series describing the evolution of the game variables during 10 pieces), and the solution of the case contains the profile of the player that played that game (computed from its final score). When a player starts a new game, we wait some time to construct her first performance window and then we use it as a query to retrieve the $k = 10$ most similar cases in the case base. We predict the profile of the current player by majority vote among the retrieved cases. Next, we resume our approach, see [27] for a more detailed description of this component.

When we retrieve the most similar cases to the current performance window, we only consider cases collected at the same time (piece number) in previous games because two very similar boards in different moments of the game can have very different implications regarding the player's skill level. For example, a game board with pieces only in the first rows and a few holes is normal at the beginning of the game but only expert players can have that same board configuration when the game is well advanced and the pieces fall with great speed.

The similarity between a query and a case is calculated as a linear combination of the similarities between their time series.

$$\begin{aligned} sim_c(c_1, c_2) \quad = \quad & \alpha_1 \; sim_{ts}(c_1.score, c_2.score) + \\ & \alpha_2 \; sim_{ts}(c_1.holes, c_2.holes) + \\ & \alpha_3 \; sim_{ts}(c_1.height, c_2.height) \end{aligned}$$

where the weights α_1, α_2, α_3 can be adjusted to give more or less importance to each time series. The most important one in Tetris to predict the player skill level is the game score ($\alpha_1 = 0.70$), followed by the number of holes ($\alpha_2 = 0.25$) and finally the board height ($\alpha_3 = 0.05$). These optimal weights were calculated using grid search with increments of 0.05 and 10-fold cross validation to improve the accuracy of the classifier.

We compute the similarity between 2 time series using a similarity based on the Euclidean distance [28]:

$$sim_{ts}(r, s) \quad = \quad 1 - \sqrt{\sum_{i=1}^{n} (r_i - s_i)^2}$$

where r and s are time series of size n.

Finally, inconsistent player behavior during the game can produce frequent changes in the profile prediction, producing continuous changes in the difficulty of the game. For example, an expert player may be distracted at some point in the game and appear to be a beginner, or a novice player may have a good run for a while. Sudden changes in game difficulty are often perceived as "something strange is going on" and provoke a distrustful attitude in players. In addition, the profile predictions of this module can fail (the predictions become more accurate as the game progresses), so it is advisable to use some *inertia function* that avoids transitioning between extreme profiles. In our implementation, we use a inertia function that only allows transitions between adjacent profiles. That is, if a player was classified as a beginner in the previous performance window, she can only be classified as a beginner or intermediate in the current performance window, no matter how well she places the last pieces.

C. Difficulty Adjustment

The purpose of this component is to decide *when* and *how* to modify the difficulty of the game. We focus on 2 of the 9 dimensions that represent the optimal psychological state of flow: challenge-skill balance and sense of control. This way, the decision of whether to modify the difficulty of the game and with what intensity to do so depends on the player's profile and whether she is in control within her profile.

1) Is the player in control?: The sense of *control* indicates whether an individual feels capable of performing the task because it is at an appropriate difficulty level and has been identified as one of the most important components of the flow experience [11]. If the task is too difficult, the subject experiences anxiety, or if it is too easy, the subject experiences boredom, so the feeling of control is somewhere between those two extremes. To indirectly measure the control of the player, we compute the complexity of the game state and then compare it with the mean complexities of the *good*, *too easy* and *too hard* games in our collection of labelled games from players within the same profile than the current player. For the player to be in control, that is, performing a task with the appropriate difficulty level, the complexity of the board during the last performance window should be more similar to the mean complexity of the good games than to the too easy or too hard ones. As in the previous component, we only compare the complexity of performance windows corresponding to the same moment (number of piece) in different games.

We compute the *game state complexity* after each piece is settled in it final location with the following formula:

$$C = \overline{a}^2 + \sigma(contour) + h$$

where $\overline{a}$ is the median height of the columns on the board squared, $\sigma(contour)$ is the standard deviation of the height of the columns, and h is the number of holes between the pieces. These variables represent challenges the player must overcome as the speed of the falling piece increases: low board height, homogeneous board contour and few holes between the pieces.

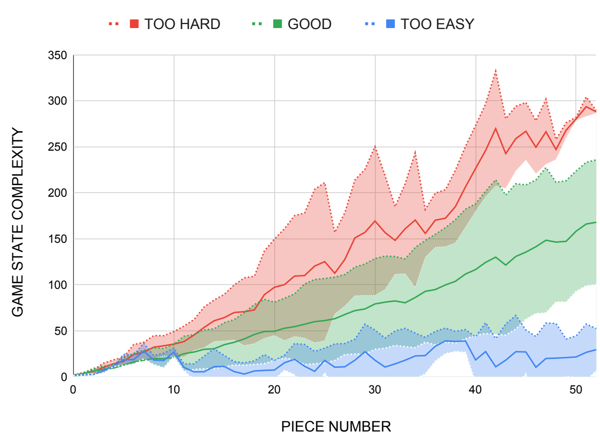

Fig. 4. Mean game state complexity and its standard deviation for beginner players as the game progresses.

Figure 4 shows the mean game state complexity and its standard deviation for players with a beginner profile as the game progresses. The red, green and blue lines represent the mean complexity for the *too hard*, *good* and *too easy* games respectively. As the game progresses, the complexities of the games come apart and each one of them increases with a different speed. This different trajectories also exists within the average players but the lines are not so far apart.

Finally, to decide if the player is in control or not (a binary variable), we compare the complexity of each game state in the last performance window with the mean complexity of the *good*, *too easy* and *too hard* games in the same instant. The player will be in control when most of those complexities are closer to the good games than to the too easy or too hard games. If the player is not in control, the difficulty of the game is not at the appropriate level for her current skill level.

2) Modify the difficulty in Tetris: The difficulty in Tetris depends basically on two variables: the type of pieces delivered and the speed at which it falls. Using these two variables, we can make the game more or less difficult and adapt it to the skill level of the current player.

The next piece is probably the most important event in the Tetris game. A good piece that fits the current configuration of the board can potentially reduce the board height, while a bad piece could greatly increment the complexity of the board. We assess how good each piece is for the current board with a heuristic function that evaluates each game board that can be reached by placing that piece in each possible final location. The heuristic value associated with each game board is the board height plus the number of holes, so the smaller the value the better the board. The heuristic value of each piece is the best (lowest) value of the reachable boards (so we assume the player will place it optimally). It is relevant to mention that it might not be a good idea to always provide the best or worst next piece, because some of them might be selected too frequently and be perceived negatively by the players. It is better to select one of the best or worst pieces randomly to ensure enough variability.

The difficulty of the game can also be changed by altering the speed at which the pieces fall. But we must do this carefully, because the falling speed is supposed to always increase in Tetris, and we don't want players to detect that we are changing the game. Also, if we reduce the drop speed too much, the game might never end. Therefore, the correct approach in this case is to slightly increase or decrease the drop speed increment each time the player places another piece.

Although our methodology can be used to make the game more or less difficult, in the experiment described next we have focused solely on making the game easier and thereby helping less experienced players. We think that beginner players have more trouble enjoying the game and getting into flow, while expert players already have the necessary skills to achieve the proposed challenges and need little or no help to have a satisfying experience. Therefore, we define two different adaptation's levels to produce a challenge-skill balance that consider both the player profile and the feeling of control:

- *Speed-piece adaptation* This level modifies the falling speed and the next piece. The next piece is selected randomly among the best 3 candidates. The falling speed rate increases by half of what it normally would. This adaptation level is active when the player is a beginner and is not in control.
- *Piece-only adaptation* This level modifies only the next piece. This adaptation level is active when either the player is a beginner and is in control, or when the player is average and is not in control.

IV. Experiment

In the experiment we compare three different versions of Tetris: normal, trivial and balanced (DDA). The *normal* version corresponds to the original Tetris Analytics system in which the next piece is chosen randomly, while the other two versions are modifications of this same system. The *normal* version is the original Tetris game without any modifications. The *trivial* version always returns a good next piece so that the player can easily make lines. The *balanced* version analyses the player's performance and changes the difficulty level accordingly as we have explained in the previous section. The goal of this experiment is to measure the flow that players experience during the games.

The experiment will test the following hypotheses:

- H1: We get more players to experience flow using DDA (balanced version) than without it (normal version).
- H2: It is harder for beginners to experience flow compared to intermediate or advanced level players (normal version).
- H3: If we provide an easy Tetris version to players with a low skill level, then they will experience flow (trivial version).
- H4: If Tetris is too easy, then the players will not experience flow (trivial version).

A. Experimental setup

The experiment was performed on site with the presence of a moderator that provided the initial instructions and guided the experiment. 40 volunteers participated in the experiment (computer science students other than those who participated in Section III-A1). They were asked to fill out a form with basic demographic information (*age*, *sex*) and answer some questions regarding their use of video games: *Are you a regular player?* (yes/no), *How many hours do you play per week?* (0-5, 5-10, 10-15, 15-20, +20), *Do you know Tetris?* (yes/no), *What skill level do you have in Tetris?* (never played before, beginner, intermediate, advanced).

Each participant began by playing Tetris for 40 seconds to become familiar with the controls and to remember how the game works. Next, each participant played 3 complete games, filling out the SHORT Flow State Scale (S-FSS) [12] questionnaire after each game. Finally, each participant chose the game she liked the most (1, 2 or 3). The experiment was designed to last approximately 30 minutes.

The participants did not know the purpose of the experiment or whether the games corresponded to different versions of Tetris. Participants were randomly divided in 3 groups so that they played the different versions in different order:

- Group 1: trivial, balanced, normal.
- Group 2: normal, trivial, balanced.
- Group 3: balanced, normal, trivial.

The SHORT Flow State Scale (S-FSS) [12] determines whether a player has entered flow during an activity (Table I). It contains nine items, and the scores of each one represents a dimension of flow. The scale is designed as a post-event assessment of flow, with instructions written to connect the subject to a recently completed activity. A more accurate evaluation of the flow state is possible when the scale is administered close to the end of the game. Responses range from 1 (strongly disagree) to 5 (strongly agree). A low response value indicates that the subject's experience was not substantially of a flow nature. Conversely, a high response value symbolizes that the individual experienced flow substantially. The average score of 3 on the status scales represents a choice of "neither agree nor disagree". This average score may indicate some degree of approval, but it may also mean some ambiguity regarding the item's relevance to the person's experience. Nevertheless, it is reasonable to interpret moderate level scores as neither strong evidence that the person has experienced flow nor strong proof that the person's experience did not include the flow.

B. Results

The age of the 40 participants ranged from 18 to 42 years, with a mean age of 27.3 years. More than half of the participants played video games more than 5 hours per week (see Figure 5). Also, more than half of the participants considered themselves beginners (see Figure 6).

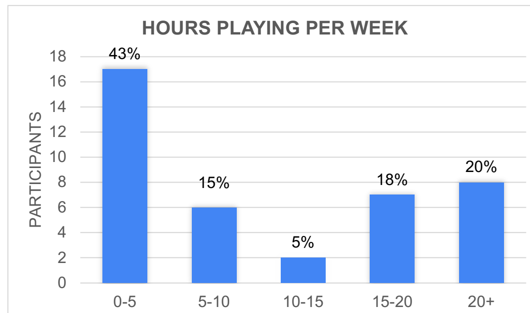

Fig. 5. Hours per week that participants play video games.

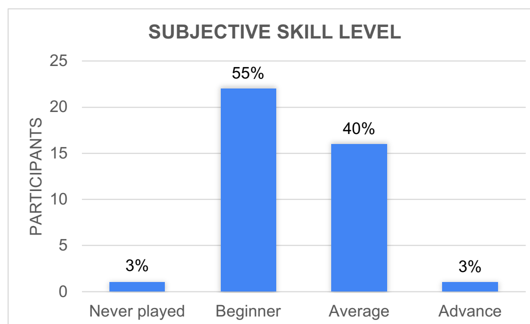

Fig. 6. Subjective skill level of the participants.

Let's start the analysis with the participants' favorite version of Tetris. In the last question of the experiment, participants chose which of the 3 games they liked the most. Figure 7 shows the percentage of times each version was chosen: 20% of the players selected the normal version, 27.5% selected the trivial version, and 52.5% selected the balanced version. If we analyze the results filtered by the player's subjective skill level of Tetris (the level they believed they had in the initial questionnaire), the balanced version was chosen twice as often as the next version (trivial) by both beginners and intermediate players. The advanced player also chose the balanced version, and the player who had never played Tetris before chose the trivial version.

The main goal of the experiment was to determine which version of Tetris generates the highest flow state. Since there were 40 participants and each of them played the 3 versions of Tetris, we had 40 SHORT Flow State Scale questionnaires for each version of Tetris. We compare Tetris versions in pairs in order to determine whether a version of Tetris produce a

TABLE I
EACH ITEM FROM THE SHORT FLOW STATE SCALE AND THE DIMENSION IT REPRESENTS [12].

#	Dimension	Sentence
1	Challenge-Skill Balance	I felt I was competent enough to meet the demands of the situation
2	Merging of Action and Awareness	I did things spontaneously and automatically without having to think
3	Clear Goals	I had a strong sense of what I wanted to do
4	Unambiguous Feedback	had a good idea about how well I was doing while I was involved in the task/activity
5	Concentration on the Task at Hand	I was completely focused on the task at hand
6	Sense of Control	I had a feeling of total control over what I was doing
7	Loss of Self-Consciousness	I was not worried about what others may have been thinking of me
8	Transformation of Time	The way time passed seemed to be different from normal
9	Autotelic Experience	I found the experience extremely rewarding

TABLE II
AVERAGE FLOW STATE PER TETRIS VERSION AND P-VALUE.

	Average Flow			**p-value**		
	normal	**balanced**	**trivial**	**normal vs balanced**	**balanced vs trivial**	**normal vs trivial**
Flow score	3.778	4.000	4.003	**0.001**	0.973	0.033

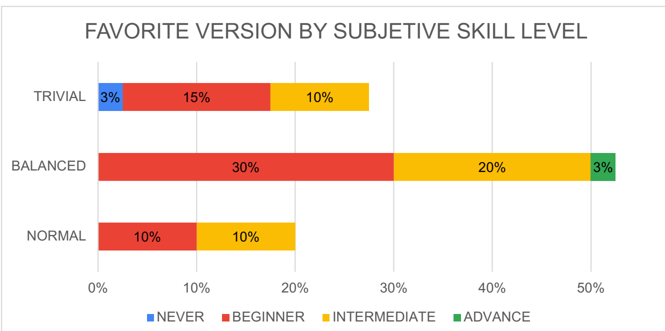

Fig. 7. Favorite version filtered by subjective skill level.

statistically significant higher flow state than other, by running a paired t-test for every considered pair of versions: normal vs. balanced; balanced vs. trivial; normal vs. trivial. Since we are running 3 different comparisons for the same dataset, we apply Bonferroni correction [29] and instead of considering a Type I error rate of 5% [30] we use a significance threshold of 0.017% ($\approx 0.05/3$).

As shown in Table II, the average flow score of the balanced version (M=4, SE=0.602) is significantly higher than the average flow score of the normal version (M=3.778, SE=0.54) with $p < .017$. Therefore, we accept *H1: we get more players to experience flow using DDA*.

To test *H2: it is harder for beginners to experience flow compared to intermediate or advanced level players*, we grouped participants according to their actual skill level (in contrast to the subjective skill level used before) using their final scores in the normal Tetris games. 21 participants were classified as beginners and 19 as intermediate players. We found no significant differences between the average flow scores in both groups of players, so we reject *H2*. However, if we separate the data and run again the paired t-test for beginners (Table III) and intermediate players (Table IV) separately, we get additional insights. For beginners, both the balanced and trivial versions provide significantly ($p < .017$) higher flow than the normal version, while for the intermediate players it is only the case with the balanced version.

To test *H3: If we provide an easy Tetris version to players with a low skill level, then they will experience flow* we compared the trivial and normal versions of Tetris within the group of beginner players. Both the balanced and trivial versions of Tetris are easier than normal Tetris. The balanced version adapts the challenges to make them easier when necessary, while the trivial version always provides help. The average flow score of the trivial version (M=4.037, SE=0.628) is significantly higher than the average flow score of the normal version (M=3.778, SE=0.415) with $p < .017$. These results show that both versions providing easier challenges helped beginners to experience flow more strongly than in normal Tetris. Therefore, we accept *H3*.

Finally, we accept *H4: if Tetris is too trivial, then the players will not experience flow*, because the resulting flow in the trivial version is not significantly higher ($p < .017$) than the flow measured with the normal or the balanced versions in our experiment (see Table II). Nevertheless, if we consider just beginners (Table III) then flow in the trivial version is significantly higher than in the normal one, so we could reject H4 for beginners.

C. Discussion

According to our results in this experiment, we can conclude that:

- The average flow experience when playing a balanced version is significantly higher than when playing the normal version.
- The average flow experience from the balanced version is not significantly higher than the trivial version.
- For beginners, both the trivial and balanced versions improve the flow experience compared to the normal version.
- We have no evidence that it is harder for beginners to get into flow compared to intermediate and advanced players.

We expected players to find the trivial version boring compared to the balanced version, but that was not the case. This is probably due to the fact that most of the participants

TABLE III
BEGINNERS' AVERAGE FLOW SCORE PER VERSION AND P-VALUE .

Beginners						
	Average Flow			p-value		
	normal	**balanced**	**trivial**	**normal vs balanced**	**balanced vs trivial**	**normal vs trivial**
Flow score	3.788	3.979	4.037	**0.010**	0.295	**0.004**

TABLE IV
INTERMEDIATE'S AVERAGE FLOW SCORE PER VERSION AND P-VALUE.

Intermediate Players						
	Average Flow			p-value		
	normal	**balanced**	**trivial**	**normal vs balanced**	**balanced vs trivial**	**normal vs trivial**
Flow score	3.766	4.023	3.965	**0.003**	0.689	0.159

in the experiment consider themselves beginners and had not played Tetris for a very long time, so they were happy to play an easy version of the game. Besides, participants scored better on the trivial version (average score 4720) than on the balanced one (average score 3659) and that can influence the answers to the questionnaire. In addition, each game lasts about 3-5 minutes and that is probably not long enough to get bored even if the game is very easy. We believe players would find the trivial version tedious if they had to play it more than a few minutes. Furthermore, the balanced version was chosen as the favorite one by 53% of the participants so we can conclude that the balanced Tetris version provides a more rewarding experience than the trivial one.

Regarding the limitations of the study, we have identified some aspects that should be further investigated in the future. First, the 40 participants were related to the computer field, so they may not be representative of the general population of gamers. Second, our study does not consider personal characteristics that cause some people to experience flow with greater or lesser difficulty and intensity. Third, our study focuses on beginner or intermediate players, and the balanced version that we use in the experiment only decreases the difficulty of the game, never increases it. It would remain to be studied how the dynamic difficulty adjustment affects expert players who need greater challenges. Finally, our experiment only considers very short game sessions. In would be interesting to study longer periods of time to see whether the difficulty of the game evolves in line with the player's improvement.

V. CONCLUSIONS AND FUTURE WORK

This work presents a novel methodology that serves as a guide to implement DDA in video games using ideas from the theory of flow. Our methodology is divided into 2 phases: pre-analysis and online game play. During pre-analysis we collect and analyze game traces to identify the most relevant game variables to assess the performance of the player. We also create a case base of game fragments labeled with the player's skill level and a satisfaction scores. The second phase takes place during the game and continuously predicts the current player's profile and decides when and how to adapt the difficulty of the game. To predict the player's profile we compare her performance during the last performance window with those stored in the case base. In turn, the decision whether to alter the difficulty of the game depends on if the player is in control, that we measure indirectly by comparing the complexity of the last game states with the average complexity of the games in the case base that were rated as good, too easy or too difficult by other players with the same profile.

We have implemented this methodology in Tetris and performed an experiment that shows that the average flow experience is significantly higher using DDA that in the normal version. We found no overall significant difference between the flow experienced between the DDA and trivial versions. We think this is because the participants were mostly beginners with little experience playing Tetris and only played one trivial game that lasted just 3 or 4 minutes and that was not enough to find it tedious. Nevertheless, differences between trivial and balanced versions can be found if we study beginners and intermediate players separately. While we found a significant difference for both versions with respect to the normal one in beginners, for intermediate players we only found the balanced version to be better than the normal one.

Although this work focuses on a particular game, we believe that the proposed methodology is general enough to be applied to a wide range of games. The main decisions that must be made to use this methodology in a particular game are (1) to identify the most relevant game variables to assess the performance of the player during the game, (2) to decide the length of the performance windows used for DDA, (3) to define a function to measure the complexity of the game during a performance window, and (4) to decide how to make the game more difficult or easier depending on the player's needs at any given time. For example, in a classic game such as Space Invaders the player's performance could be measured using variables such as the game score, the number of ships and aliens destroyed, the number of shots that did not hit any enemy, ...; new performance windows could be created every 10, 20 or 30 seconds depending on how often we want to reevaluate the difficulty of the game, with values collected every few seconds; the complexity of the game could be measure using the speed of the aliens, their distance to the ground, the amount of barriers remaining for the player to cover, ...; and the difficultly of the game could be changed by altering the speed of the aliens or their fire rate. Of course, these are just ideas that should be tested and evaluated by both game designers, who know in detail how the different game mechanics work, and data analysts who can find and verify patterns in the game traces.

Our proposal has some limitations as well. We use a case-based implementation and, therefore, we require a good case base that effectively represents different types of players and games (with a good coverage and diversity of cases). In addition, the decision of using ex-post questionnaires instead of real-time sensors to measure the player experience has some advantages (it is a not intrusive approach and it does not alter the player's experience), but it also means that we only get feedback about the whole game (or at least a level of the game) so we cannot differentiate which specific parts induced more or less flow in the player. As part of the future work, we would like to research the use of non-intrusive devices to identify and label good, too easy and too hard game fragments in the case base. Note that those devices would only be needed during the construction of the case base and not to perform DDA once the game is released. The spectacular progress of smart watches and bracelets in recent years makes us feel hopeful in this regard. In fact, the use of these devices could also help us to measure the effectiveness of a difficulty adjustment by examining the subsequent biometric response of the player.

Although the methodology presented in this paper is general, we have only tested it with beginner and intermediate players making the game easier and never more difficult. We would also like to test this type of DDA during longer periods of time to see if the difficulty of the game evolves in line with the player's improvement. We are interested in the use of DDA in serious games where keeping the player engaged is critical to enhance learning over longer periods of time.

Acknowledgments

This work has been partially supported by the Spanish Committee of Science and Innovation (PID2020-114596RB-C21) and the Complutense University of Madrid (Group 921330). We would like to thank the anonymous reviewers for their helpful and insightful comments.

References

[1] A. Denisova, P. A. Cairns, C. Guckelsberger, and D. Zendle, "Measuring perceived challenge in digital games: Development & validation of the challenge originating from recent gameplay interaction scale (CORGIS)," *Int. J. Hum. Comput. Stud.*, vol. 137, p. 102383, 2020.

[2] R. J. Pagulayan, K. Keeker, D. Wixon, R. L. Romero, and T. Fuller, "User-centered design in games," in *The human-computer interaction handbook*. CRC Press, 2002, pp. 915–938.

[3] D. Charles, A. Kerr, and M. McNeill, "Player-centred game design: Player modelling and adaptive digital games," in *Proceedings of the Digital Games Research Conference*, vol. 285, 2005, pp. 285–298.

[4] 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.

[5] O. Missura and T. Gärtner, "Player modeling for intelligent difficulty adjustment," in *Discovery Science, 12th International Conference, DS 2009, Porto, Portugal, October 3-5, 2009*, 2009, pp. 197–211.

[6] A. Ram, S. Ontañón, and M. Mehta, "Artificial intelligence for adaptive computer games." in *FLAIRS Conference*, 2007, pp. 22–29.

[7] R. Lopes and R. Bidarra, "Adaptivity challenges in games and simulations: a survey," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, no. 2, pp. 85–99, 2011.

[8] J. Chen, "Flow in games (and everything else)," *Communications of the ACM*, vol. 50, no. 4, pp. 31–34, 2007.

[9] M. Csikszentmihalyi, *Flow: The psychology of optimal experience*. Harper & Row New York, 1990.

[10] A. Aamodt and E. Plaza, "Case-based reasoning: Foundational issues, methodological variations, and system approaches," *AI Commun.*, vol. 7, no. 1, pp. 39–59, 1994.

[11] J. Nakamura and M. Csikszentmihalyi, "The concept of flow," in *Flow and the foundations of positive psychology*. Springer, 2014, pp. 239–263.

[12] S. Jackson, B. Eklund, and A. Martin, "The flow manual the manual for the flow scales manual. sampler set," *Mind*, vol. 2011, pp. 1–85, 2011.

[13] C. A. Cruz and J. A. R. Uresti, "Player-centered game ai from a flow perspective: Towards a better understanding of past trends and future directions," *Entertainment Computing*, vol. 20, pp. 11–24, 2017.

[14] P. Sweetser, D. M. Johnson, P. Wyeth, A. Anwar, Y. Meng, and A. Ozdowska, "Gameflow in different game genres and platforms," *Comput. Entertain.*, vol. 15, no. 3, pp. 1:1–1:24, 2017.

[15] G. Chanel, C. Rebetez, M. Bétrancourt, and T. Pun, "Emotion assessment from physiological signals for adaptation of game difficulty," *IEEE Trans. Syst. Man Cybern. Part A*, vol. 41, no. 6, pp. 1052–1063, 2011.

[16] M. Klasen, R. Weber, T. T. Kircher, K. A. Mathiak, and K. Mathiak, "Neural contributions to flow experience during video game playing," *Social cognitive and affective neuroscience*, vol. 7, no. 4, pp. 485–495, 2012.

[17] S. Park, H. Sim, and W. Lee, "Dynamic game difficulty control by using eeg-based emotion recognition," *International Journal of Control and Automation*, vol. 7, pp. 267–272, 03 2014.

[18] A. Stein, Y. Yotam, R. Puzis, G. Shani, and M. Taieb-Maimon, "Eeg-triggered dynamic difficulty adjustment for multiplayer games," *Entertain. Comput.*, vol. 25, pp. 14–25, 2018.

[19] T. Alves, S. Gama, and F. S. Melo, "Flow adaptation in serious games for health," in *6th IEEE International Conference on Serious Games and Applications for Health, SeGAH 2018, Vienna, Austria, May 16-18, 2018*, J. L. Vilaça, T. Grechenig, D. Duque, N. F. Rodrigues, and N. Dias, Eds. IEEE, 2018, pp. 1–8.

[20] S. A. Jackson, A. J. Martin, and R. C. Eklund, "Long and short measures of flow: The construct validity of the fss-2, dfs-2, and new brief counterparts," *Journal of Sport and Exercise Psychology*, vol. 30, no. 5, pp. 561–587, 2008.

[21] K. Spiel, S. Bertel, and F. Kayali, ""not another Z piece!": Adaptive difficulty in TETRIS," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, Denver, CO, USA, May 06-11, 2017*, G. Mark, S. R. Fussell, C. Lampe, m. c. schraefel, J. P. Hourcade, C. Appert, and D. Wigdor, Eds. ACM, 2017, pp. 5126–5131.

[22] A. Nagle, P. Wolf, and R. Riener, "Towards a system of customized video game mechanics based on player personality: Relating the big five personality traits with difficulty adaptation in a first-person shooter game," *Entertain. Comput.*, vol. 13, pp. 10–24, 2016.

[23] M. P. Silva, V. do Nascimento Silva, and L. Chaimowicz, "Dynamic difficulty adjustment on MOBA games," *Entertain. Comput.*, vol. 18, pp. 103–123, 2017.

[24] J. E. Cechanowicz, C. Gutwin, S. Bateman, R. L. Mandryk, and I. Stavness, "Improving player balancing in racing games," in *Proceedings of the first ACM SIGCHI annual symposium on Computer-human interaction in play, Toronto, ON, Canada, October 19 - 22, 2014*, L. E. Nacke and T. C. N. Graham, Eds. ACM, 2014, pp. 47–56.

[25] O. Missura and T. Gärtner, "Predicting dynamic difficulty," in *Advances in Neural Information Processing Systems*, 2011, pp. 2007–2015.

[26] A. Zook and M. O. Riedl, "A temporal data-driven player model for dynamic difficulty adjustment," in *Proceedings of the Eighth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, AIIDE-12, Stanford, California, USA, October 8-12, 2012*, M. Riedl and G. Sukthankar, Eds. The AAAI Press, 2012.

[27] D. S. L. Ariza, A. A. Sánchez-Ruiz, and P. A. González-Calero, "Time series and case-based reasoning for an intelligent tetris game," in *Case-Based Reasoning Research and Development - 25th International Conference, ICCBR 2017, Trondheim, Norway, June 26-28, 2017, Proceedings*, ser. Lecture Notes in Computer Science, D. W. Aha and J. Lieber, Eds., vol. 10339. Springer, 2017, pp. 185–199.

[28] S. Rani and G. Sikka, "Recent Techniques of Clustering of Time Series Data: A Survey," *International Journal of Computer Applications*, vol. 52, no. 15, pp. 1–9, 2012.

[29] N. J. Salkind, Ed., *Encyclopedia of research design*. SAGE Publications, Inc., 2010.

[30] J. de Winter and D. Dodou, "Five-point likert items: t test versus mann-whitney-wilcoxon," *Practical Assessment, Research and Evaluation*, vol. 15, pp. 1–16, 2010.