Measuring Control to Dynamically Induce Flow in Tetris

Diana Sofía Lora Ariza, Antonio A. Sanchez Ruiz-Granados, Pedro A. Gonzalez Calero, and Irene Camps Ortueta

***Abstract*—TODO: VOLVER A ESCRIBIR**

We present a dynamic difficulty adjustment methodology in this paper. This methodology serves as a guide to implement a DDA system in a concrete video game called Tetris Analytics. We use flow theory to adapt the game's difficulty, and we focus on skill-challenge balance and sense of control. We created three versions of Tetris (trivial, balanced, and normal) and asked a group of volunteers to play the different versions and answered the Short Flow State Scale (S-FSS) after each game. We found that volunteers experience overall significantly more flow in the balanced games than in the normal games. In this case, the dimensions challenge-skill balance, merging of action-awareness, and sense of control are the strongest. Furthermore, we found that there is no significant difference between trivial and balanced games. However, the autotelic experience dimension was significantly higher in balanced games. This dimension describes the intrinsically rewarding experience an individual feels when on flow [1].

***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 dialogs, ... and specially the difficulty levels of the game that usually involve defining different sets of behaviours for the AI characters that will be used according to the player's skill level [2]. In other words, the "intelligence" seen in most video games today is the result of trying to anticipate different player behaviours and then, during the production stage, to implement a standard set of actions to respond appropriately in each expected scenario [3]. 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 [4].

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 [2]. In particular, the classical predefined difficulty levels present in most video games are not an optimal solution because the *difficulty* is not a static property but a subjective factor derived from the interaction between the player and the proposed challenge [5]. 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 [6], [7]. 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 [8]. For this reason, DDA has captured the attention of leading companies in the video game industry and, for example, Sony has patented an AI system to dynamically adjust the difficulty of Sekiro's final bosses [1] depending on the player's skill level.

This paper presents a novel implementation of a DDA system in the Tetris video game based on the *flow* theory. This theory studies the nine dimensions required to achieve an optimal experience when performing a task. Informally, we say that a person is experimenting flow when she is completely focused and engaged in a task, oblivious to the pass of time and enjoying. Previous research about flow theory in the video games context shows that not all dimensions are equally relevant to achieve flow when playing video games [9], and we focus on two of the most important ones: challenge-skill balance and sense of control. Besides, we also focus on beginner players that have very limited experience with Tetris. Our hypothesis is that beginner players might have more problems to reach flow that expert players since they do not know the game and can easily become overwhelmed with the difficulty and abandon the game. Expert players, on the other hand, keep playing the game after several days, weeks or months so we can assume that they are already having a satisfactory experience with the game. Finally, although we

M. Shell was with the Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332 USA e-mail: (see http://www.michaelshell.org/contact.html).

J. Doe and J. Doe are with Anonymous University.

Manuscript received April 19, 2005; revised August 26, 2015.

[1]https: //www.sekirothegame.com/de/

focus on the Tetris game our approach is general enough to be adapted and used in other video games.

This paper goes as follows. First, we describe the flow theory, its dimensions, the emotions that might arise when doing a task and how to measure it. Second, we present the related work on how other researchers have attempted to employ DDA and the flow theory in video games. Third, we explain our approach to DDA based on flow and how we have implemented its different components. We use a case-based reasoning approach to dynamically predict the player's skill level and adapt the difficulty of the game depending on a measure of control based on the relatively complexity of the game state when we compare it with a set of previous labeled games. Then, we describe an experiment that shows that our approach effectively improves the flow experience of the players. The paper ends with some discussions and final conclusions.

II. Flow Theory

Czikszentmihalyi [10] 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, and separately, constitute the conceptual elements of flow [1]. These nine dimensions are 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 (figure 1). 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, their skills increase and thus the challenge must grow in difficulty along with the person to keep them in flow.

Figure 2 shows 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 intensified when the challenges and skills required to perform the task are beyond the player's average levels. Furthermore, the challenges presented in a task should grow in difficulty as the person masters their skills. Otherwise, the task ceases to be enjoyable. This is why it is so difficult to achieve and maintain that state of balance over time.

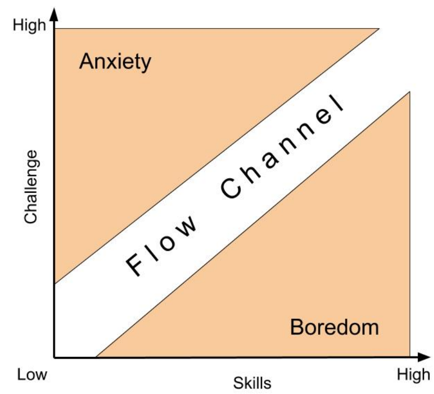

Fig. 1. The original flow model [11]. The figure shows the importance of the balance between the challenge and the person's abilities. If the challenge is greater than the person's abilities, then the person experiences anxiety. Conversely, if the challenge is too easy, the person experiences boredom when performing the task.

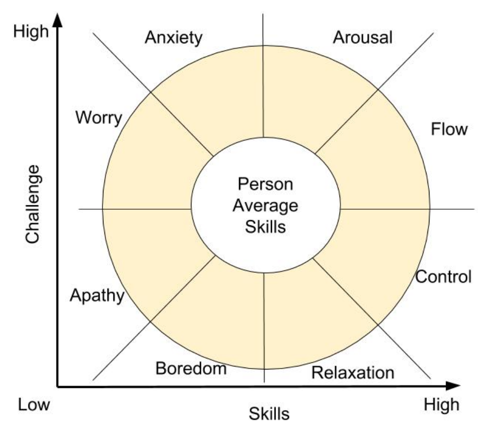

Fig. 2. The eight emotions a person can experience while performing a task [11].

The SHORT Flow State Scale (S-FSS) [1] determines whether a player has entered flow during an activity (table I). It contains nine items, and the scores of each one represent 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 symbolises that the individual experienced a substantially fluid experience. 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 of the item. However, 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.

TABLE I
EACH ITEM FROM THE SHORT FLOW STATE SCALE AND THE DIMENSION IT REPRESENTS [1].

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

III. RELATED WORK

There is an increasing interest in how dynamic difficulty adjustment improves user experience. Researchers are using machine learning techniques to adjust the game based on players performance to create complex behaviours [12], [13], [14], [15], [16]. Lopes and Bidarra [17] surveyed the state of art of adaptivity in games and simulations, from both academia and industry. They concluded that games and simulations adaptivity is establishing itself as a rapidly maturing field and that the advances show good results in adapting to an optimal challenge level and active states like fun, frustration, predictability, anxiety or boredom.

One example of how to implement DDA in video games is Missura and Gärtner [18] work. They used a simple game where the player shoots down alien spaceships while those shoot back. They aimed to employ dynamic difficulty adjustments by grouping players into different profiles and supervised prediction from short traces of gameplay. Each game had a limit of 100 seconds. The first 30 seconds were used to acquire data, and the rest of the game, they adjust the aliens' spaceships speed based on the player's performance.

One approach to DDA in video games is flow theory. It can keep the player in a balanced state so that the challenges in the game evolve simultaneously with the player's abilities. Descriptions of the flow experience are identical to those experienced by players when they immerse in games, such as time loss and external pressure, among others [19]. Plus, gamers value video games that provide a flow experience [20].

Currently, there is a division among researchers because some believe that all requirements must occur to enter flow and others believe that not all are necessary [21]. Cairns et al. [22] have suggested that flow is an "all or nothing" experience, during which the individual must meet all the criteria to enter flow. Otherwise, the player will not enter flow. Other researchers have argued that flow does not need to fulfil all criteria simultaneously [23], [24], [19], [25], [26]. One example is Klasen et al. [9], who assessed how the brain responds to the different dimensions that contribute to the flow experience in a video game. The subjects played *Tactical Ops: Assault on Terror*, a first-person shooter game, while an MRI machine captured images of their brain activity. The data showed that the balance between challenge and skill, sense of control and concentration are most representative of the flow experience.

Additionally, Cruz et al. [26] made a literature review of flow theory and the different attempts to adapt it in the context of video games. One approach is to relate flow antecedents to game attributes. Regarding this matter, Jones mapped flow dimensions to game's features to create more engaging computer-based learning environments (CBLEs) for all types of learners. He first conducted empirical research to find out which game features promote player engagement [27]. His hypothesis was that knowledge of how video games engage players could be extrapolated and used in CBLEs environments. The results of this study led the author to relate the similarity between flow theory and game design methodologies. Jones subsequently presents a mapping between eight dimensions of flow and game features [28]. In addition, Cruz et al. make a comparative analysis between the work of Cowley et al. [29] and Pavlas [30]. Both present a formal mapping between flow elements with game features and describe the interactions between players and a flow-based video game. According to Cruz et al., both models complement each other. Pavlas focuses on how the elements of flow concepts are related, which facilitates the understanding and visualisation of how a player-centred game design should be when using flow theory. While Cowley et al. provide a better idea of the player experience and how we should adapt game entities to foster flow.

IV. DDA IN TETRIS

We created a methodology for dynamic difficulty adjustment in video games that serves as a guide for implementing a DDA system during the video game development process. Its successful implementation improves the player's experience by balancing their real skills and the skills required to complete the proposed challenge. The methodology is composed of 3 main phases: *players' pre-analysis*, *players' categorisation*, and *difficulty adjustment*. Figure 3 shows its main components and how they relate to each other. The first one is an "offline" phase where we gather data and analyse it to learn more about the game, its variables, and how players interact with the game. Then, we implement the "online" phases in the game. These are executed one after the other continuously during the game session. During *player categorisation*, we predict the skill level of the player in a current game and then, we make the necessary *difficulty adjustment* based on those results. At the beginning of the game, we wait a reasonable time to collect enough data and analyse the evolution of the game variables. Then, we start the cycle of *players' categorisation* and *difficulty adjustment*, which repeats from time to time.

Below we explain how we implemented each phase in Tetris Analytics.

A. Players' Pre-Analysis

The player's pre-analysis goal is to learn more about the video game, how game variables behave and how players interact with the game. To do this, we collect a set of games and use statistical, visualisation and machine learning techniques to analyse them.

1) Game Set and Player's Profile: We have 156 games in our game set at this point, and we classify each game by player's profile and satisfaction. To distinguish the player's profile once the game ended, we must identify one or multiple variables to create a numerical representation of the players' skill level. One excellent candidate in Tetris is the total score [31]. So we use it to create the profiles and distinguish games by players' skill level. Here the profiles:

- Beginner: Games with total score between 0-2999.
- Average: Games with total score between 3000-5999.
- Expert: Games with total score greater than 6000.

Furthermore, players' satisfaction is relevant for filtering out games that were positively rated, analyse variable's behaviours, and replicate these behaviours in the future. Or on the contrary, to identify variable's behaviours that can generate a bad experience and avoid them in the future. Therefore, we ask the subject to evaluate the game once it ends with a simple and open sentence: *was a good Tetris game*. The subject should answer with a 5-value Likert scale, where 1 means strongly disagrees, and 5 strongly agrees. We consider as a *good game* only those with a score of 4 or 5. Games with a score of 1 or 2 are *bad games*. Additionally, we identify which games, in the subset of the bad game, were *hard* or *easy*.

2) Feature Selection: When a piece appears on top of the board, the player must analyse his current board and decide where he wants the piece to end up. We call the decision of what the final location of the piece will be a *tactical decision*. All the intermediate movements that the player must make to bring that piece from the top of the board to the desired position are called *technical decisions*. In our work, we use the *tactical decisions* to determine the player's performance.

We also identified which variables allow us to predict a player's profile during the game. Variable selection is a process of trial and error, where a game expert should recognise it. So, we selected those variables that we believed were good candidates to predict a player's skill level. The selected variables were:

- *Piece number* from the start of the game. As we only take into account tactical decisions, the current piece number represents the point in the game when the sample is drawn.
- *Current score* accumulated by the player once the current piece is placed in its final position.
- *Max piece height* or the highest row occupied by a piece. The maximum board height is 20.
- *Holes* or spaces under other pieces. A *hole* is a cell in the matrix that is completely covered by another part.

Figure 4 shows an example of how game features evolve in Figure 4 shows an example of how game features evolve in a game where the player is average. This player was able to settle 64 pieces before the game finishes and got a total score of 2480. The yellow line describes the score evolution that grows slightly with each new piece and more abruptly when the player makes a new line. The red line represents the max piece height of the board. This variable ranges between 1 and 20. When it reaches its max value mean that the player loses. In the first 30 tactical decisions, the player plays in the lower half of the board. Then, the max piece height grows fast because the pieces fall too quickly for the players to set them compactly. Similarly, the blue line represents the number of holes which also rises after the 25th tactical decision.

Time has a key role in the performance analysis because we can evaluate the evolution of the player's skills in a given time range by organising the variables in chronological order. A game variable represents a time series containing data collected in chronological order. Thus, the collected variables for evaluating the player's performance are called a *performance window*. In other words, a *performance window* is the time series of the most recent game variables collected for analysis. It is a matrix $mtimesn$, where m is the number of variables selected and n is the length of the time series.

3) Measuring Flow: A simple way to find out details like the player's preferences and their experience during the game is with questionnaires before or after the games. We use the SHORT Flow State Scale (S-FSS) [1] to determine whether a player has entered flow during the game. The advantage of using these scales is that we can not only get an overall flow score, but we also get a score for each flow dimension. This is particularly interesting when we want to validate whether the changes made in a game affect a specific dimension that we consider to be a priority.

B. Player's Categorisation

We use the cased-base reasoning (CBR) to decide the skill level of a new player during the game. The first step in this phase is to build a case base that the CBR system will use to predict the player's skill level. A case is a collection of time series that describes the evolution of the selected features in a time range and is labelled with the player's profile. Once the case base is ready, the game waits until having enough data from the player and builds a query out of the *performance window* data. Then, retrieves the most similar cases to the query in the case base and determines which label (player's profile) corresponds to the query. A majority of occurrences among the retrieved cases determines the player's skill level. It is important to mention that we only consider cases at the same time in previous games when searching for similar ones in the case base. Mainly because we want to predict the player's skill level by looking for other games where the player has made similar decisions, and by the logic of Tetris, we cannot compare the skill level a player has between time 1-10 with the skill level a player has between time 40-50. One reason is that the speed at which the piece falls increases and is different in these two moments. At the beginning of the game, it falls

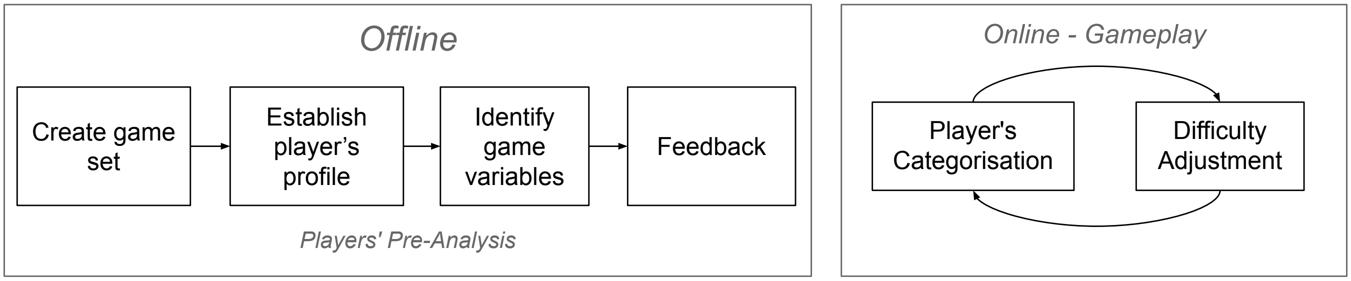

Fig. 3. DDA methodology's phases: players' pre-analysis, players' categorisation, and difficulty adjustment.

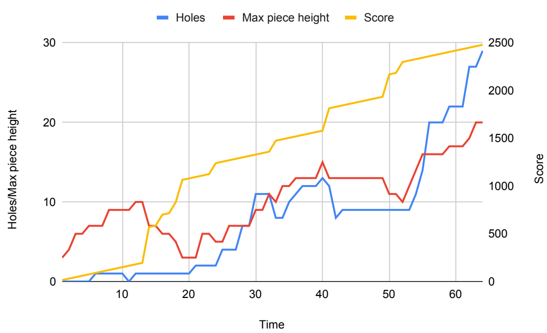

Fig. 4. Feature evolution of an average player.

slower, and the player has more time to analyse the board and decide where to place it. Although, it drops much faster when the game is in the middle or at the end. A piece's falling speed influences the player abilities to locate it where they want. Another reason is that we are evaluating time series, and in this data type, time is an important variable. For example, a player's skill level who has placed 50 pieces and has low piece heights and few gaps is not the same as a player who has just started the game and therefore has low piece heights and few holes. Even though the board may look similar, in the first example the player is good, while in the other example the player has only just started the game.

The similarity between two cases was calculated as a linear combination of the similarities between their time series.

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

where the weights $alpha_1$, $alpha_2$, $alpha_3$ can be adjusted to give more or less importance to each variable and n is the size of the time series containing the evolution of the player in that time interval. The most important variable in the similarity measure for this configuration is the accumulative game score ($alpha_1 = 0.70$), followed by the number of spaces ($alpha_2 = 0.25$) and finally the board height ($alpha_3 = 0.05$). These weights were calculated using all possible combinations with 0.05 increments in each of the weights.

To compare time series, we used a simple similarity measure based on the Euclidean distance [32]:

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

where we have two time series r and s of size n, and compare the values at the same moment i.

Additionally, we must consider that player's behaviour in a game is unpredictable. Even an expert may distract during a game due to external reasons and may look like a beginner. Or the opposite case, the novice is starting to master the game and luckily has had a good run. In neither case should the player's prediction jump from one extreme profile to another. The changes to the game's difficulty are base on the player's predicted profile. And we should avoid having abrupt changes in the game's behaviour that the player may perceive as strange. Also, the CBR classifier has an accuracy of 71.98%, so it is not perfect, and smooth difficulty changes can partially hide those problems. To this end, we added an inertia function that eases the transition between extreme profiles, e.g. from novice to expert and from expert to novice. This inertia function is applied after the profile prediction and compares the predicted profile in the previous tactical decision with the predicted profile in the current tactical decision.

C. *Difficulty Adjustment*

DDA focuses on analysing the player's skill level and adjusting the game's difficulty at a suitable level. We focus solely on making the game easier for players who struggle with the game. We believe that players who don't have the required skills have a tougher time enjoying the game and getting into flow. Whereas experts already have the skills to achieve the proposed challenges and need little or no help. So our goal is to make the game easier (if necessary) and not harder. Therefore, we want to answer the questions a) when to modify the difficulty in Tetris and b) how to modify it.

1) When to modify the difficulty in Tetris: We must consider the selected approach, *flow theory*, to modify the difficulty in Tetris Analytics. Flow theory lists 9 dimensions that represent the optimal psychological state of flow [11]. And in this work, we focus on sense of control and challenge-skill balance to induce flow.

Control is a state where the individual feels capable of performing the task because it is at an appropriate difficulty level. If the task is too difficult, the subject experiences anxiety, or if it is too easy, the subject experiences boredom. Therefore, control is in the middle of a scale where the extremes are anxiety and boredom. So, we wanted to build a function that behaves similarly. It should take in the game's features that represent challenges to overcome and resolve the *game state complexity* at a given time. At its lowest level means boredom and at its highest represents anxiety.

We want to determine the *game state complexity* in the latest performance window to identify if a player is in control. So, we considered the challenges the player must overcome while the speed of the falling piece increases. Therefore, we examine variables that respond to these challenges, like reducing the piece's height, homogenising the contour of the board, and minimising gaps. Piece's height reduction is one of the challenges and is a variable that has a visual impact on the player. When the game starts, the player has some peace of mind because the board is empty. Players must make the right choices (make lines or keep the pieces compact) to maintain the board clear. The speed at which the pieces fall increases as the game progresses, so having a low piece's height gives the player more time to analyse the possible options where to place the falling piece. That is why this variable becomes more relevant as time goes by. A second challenge is to have a homogeneous board contour considering few variations increases the possibility to make multiple lines with a single tactical decision. Similar to Romdhane and Lamontagne [33], we want to reduce the problem space (the entire board) to just the contour of the pieces on the board and use the standard deviation of the height of each column on the board. Thus, we reduce the analysis of the variation of the board contour to a single value. Another challenge is the gaps between the pieces, as they prevent players from making multiple lines in a single tactical decision. The ability to overcome all these challenges can be the difference between having a few more moments to maximise the score or losing.

The equation we have defined to calculate the *game state complexity* for each tactical decision is:

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

where $\overline{a}$ is the median height of the pieces on the board squared, $\sigma(contour)$ is the height's standard deviation of each column on the board, and h is the number of gaps between the pieces. The complexity should be at its lowest value when the game starts and as time goes on, it increases. The complexity variable is proportional to the anxiety the player may feel at any given moment. So when a player starts a game, the complexity value is at its minimum because the board is empty, then as time goes by, the complexity increases until the player loses.

We calculate the *game state complexity* of each tactical decision in the performance window to find out if a player is in control. We compare the *game state complexity* evolution of the current game against the average complexity of the different subsets of our game set. The idea is that the variable's behaviour of the current game is similar to the *good games* and move away from the *hard bad games*. Then during the game, we compare the complexity evolution in the performance window to the average complexity of each group and identify which subset of games resembles the most. If the complexity of a tactical decision is closer to the mean of the *good games*, then we assume that the player is in control. But if the complexity of a tactical decision is closer to the average of the *hard* or *easy* games, then the tactical decision is not in control. We consider a player to have control when more than half of the tactical decisions complexity in the latest performance window are closer to the *good games* complexity.

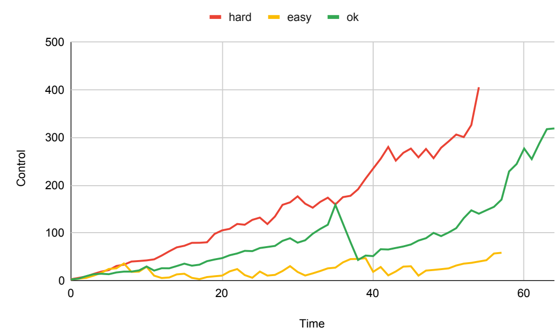

Fig. 5. Beginner's mean control behaviour per subset.

Figure 5 shows the beginner's mean control behaviour of easy, hard, and good subsets. The red line represents the mean control behaviour of the *hard games*, the green line represents the average control behaviour of the *good games* and the yellow line is the average control behaviour of the *easy games*. Figures shows the three lines are close together when the game starts, however around tactical decision 10 this changes and each takes a different direction. In general, the average players' means are closer together than the beginners. This is expected because averages have more experience and therefore they tend to have more control even in bad situations than beginners.

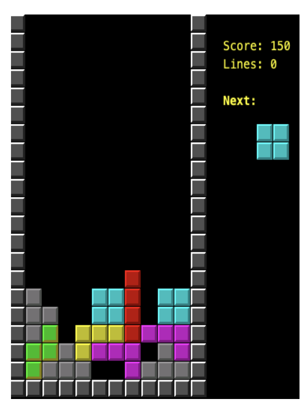

Fig. 6. Beginner's game state complexity at tactical decision 10.

Next, we present two examples showing how to determine the *game state complexity*. Figure 6 shows a game at tactical decision 10, where the player's predicted profile is beginner and the complexity is $10,25$ ($\overline{a} = 6$, $\sigma(contour) = 1,25$,

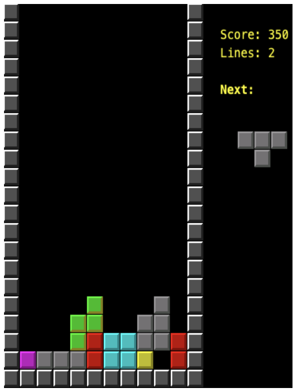

Fig. 7. Intermediate's game state complexity at tactical decision 10.

and $h = 3$). We compare the current complexity to the average *game state complexity* of *good games* and *hard games*. We do so by calculating the *game state complexity* of each tactical decision in the performance window. And then, calculate the difference of each tactical decision to the average *game state complexity* of *good games* and *hard games* at the same time. We find that the player is not in control because seven tactical decisions in the performance window have a similar complexity to the hard games. Figure 7 presents a game at tactical decision 10, where the player's predicted profile is intermediate and the complexity is $4,99$ ($\overline{a} = 3$, $\sigma(contour) = 0,99$, and $h = 1$). We find that six tactical decisions have a complexity similar to the one in the good games. Therefore, the player is in control.

2) How to modify the difficulty in Tetris: By examining Tetris's variables, we determined that the difficulty depends mainly on two variables: the type of pieces delivered and the speed at which it falls. Our first option to make the game easier is to deliver an adequate next piece, so the player makes lines. A player can potentially reduce the board's height when placing the current piece in a good spot. On that account, we pick the next piece based on the current game state. When the system decides to help the player, it checks how good each piece type is on the current board B. The number of holes is subtracted minus the max piece height to evaluate a board. The max piece height of the board is the northernmost occupied cell on the board. The number of gaps is the sum of empty cells that have at least one occupied cell above it. The lower the score on the board, the better the new possible status. When we have all the scores of the boards, we know if the player's move is among the best possible moves. Then, we select one of the best three pieces randomly. So, we evaluate each board B' generated by placing each piece type in all possible positions (rotations and columns). Each feasible location generates a new B' board, and then we arrange the B' boards from most to least favourable to the player. It is relevant to mention that our first approach was to provide the best piece always, but it turns out that this was not such a good idea. Some pieces, like the square or the I, are good candidates and are selected frequently. So, it was noticeable that the game was cheating, and the players perceived it negatively.

Another option to facilitate Tetris is by altering the falling speed of the pieces. This approach is complex because the falling speed always increases in Tetris. If done incorrectly, we could generate negative feelings in the player because they don't understand what the game is doing. Also, if we decide to reduce the falling piece's speed, the game could never end. Therefore, we reduce the rate that this variable increases each time a player places a new piece. This small change may allow the player to locate a few more pieces on the board.

We create different adaptation's levels to produce a challenge-skill balance. These try to improve the player's experience by considering all possible variations of profile and control. The fewer skills a player has and the less control they have over the game. Therefore, the more variables the adaptation level should adjust that can improve their performance. The adaptation levels are:

- *Extreme* This level modifies the standard increase in piece drop rate and the next piece that is given to the player. The drop rate increases by half of what it normally would. This adaptation level is active when the following two conditions are met 1) the player is a novice, 2) the player is not in control.
- *Normal* This level modifies only the next piece that is given to the player. This adaptation level is active when the following two conditions are met 1) the player is a beginner and is in control, 2) the player is average and is not in control.
- *Null* This level does not modify any variable.

V. Experiment

We created a flexible Tetris' version that can change behaviour based on startup configuration. We can generate three different behaviours: normal, trivial or balanced. The normal version behaves like a regular Tetris game. The trivial version always returns a good next piece so that the player can make lines. The balanced version analyses the player's performance (skill level and control) and changes the difficulty level accordingly. The goal of this experiment is to measure the flow that players experience during the games. That's why we introduce an trivial version because we want to compare a trivial solution with a more sophisticated solution (balanced version). In general, we believe that if we make the game too trivial, players may not experience flow. We want to accept or reject the following hypotheses:

- H1: We get more players to experience flow using DDA.
- H2: It is harder for beginners to experience flow compared to intermediate or advanced level players.
- H3: If we provide an easy Tetris version to someone with a low skill set, then they will experience flow.
- H4: If Tetris is too trivial, then the players won't experience flow.

We asked 40 volunteers to play Tetris to verify our implementation. Before playing any game, we asked the volunteers to fill out a form to learn more about them and their preferences. The additional information we requested was:

- *Age*
- *Sex*
- *Are you a regular player?*: options yes and no.

- *How many hours do you play per week?*: options: less than 5, between 5 till 10, between 10 till 15, 20 or more.
- *Do you know Tetris?*: options yes and no.
- *What skill level do you have in Tetris?*: options: never played before, beginner, intermediate, advanced.

Each volunteer played three games (one for each version). We created three groups where each volunteer played the games in a different order. And the group selection is made randomly before starting the experiment. The concrete order is:

- Group 1 played 1) trivial, 2) balanced, 3) normal.
- Group 2 played 1) normal, 2) trivial, 3) balanced.
- Group 3 played 1) balanced, 2) normal, 3) trivial.

Additionally, we asked the volunteer to answer *SHORT Flow State Scale (S-FSS)* after each game. Finally, we ask the player to answer *Which version of Tetris did you like the most?* With the options 1, 2 or 3.

VI. Results

In this experiment, we had 40 volunteers between the ages of 18 and 42, and the average age is 27.3 years. 70% of the volunteers were regular players, 20% play more than 20 hours a week, 17.5% play between 15 to 20 hours a week, 5% play for 10 to 15 hours a week, 15 % play 5 to 10 hours a week and the rest 5 or less hours a week (figure 8). Only one of the 40 volunteers did not know Tetris. The majority considered themselves beginners, 40% considered themselves intermediate level, the 2.5% had never played Tetris and the other 2.5% considered themselves advanced (figure 9). Also, 25% is part of group 1, the 57.5% is part of group 2 and the 17.5% is part of group 3.

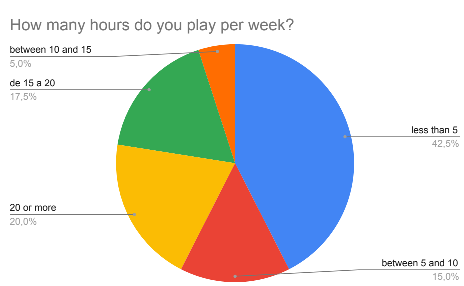

Fig. 8. Amount of hours per week that volunteers play.

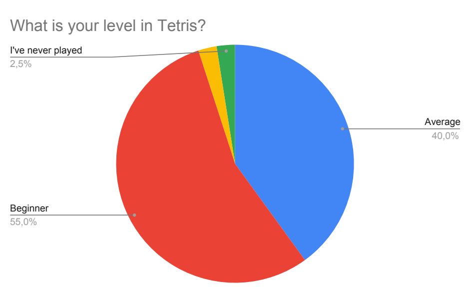

Fig. 9. Volunteers Tetris skill level.

TABLE II
Favourite version filtered by subjective skill level.

Favourite	Never played	Beginner	Intermediate	Advanced
normal	0	4	4	0
dda	0	12	8	1
trivial	1	6	4	0

First, we want to know which was the favourite version among our volunteers. 20% of players selected the normal version as their favourite, 27.5% of the players chose the trivial version as their favourite, and 52.5% picked the adjusted version as their favourite. In addition, we want to know which was the overall favourite version based on the subjective skill level of the volunteers. In the initial questionnaire, where we try to learn more about the player, we ask what level they thought they had in Tetris, and we considered this value as the player's subjective skill level. Table II shows the favourite version filtered by the subjective skill level. Only one volunteer in their 40s had never played Tetris before, and in this case, he prefers the trivial version. 22 volunteers considered themselves beginners, where the majority (54.5%) selecting that they preferred the balanced version, 27.27% preferred the trivial version and 18.18% preferred the normal version. 16 people consider that they had an intermediate skill level in Tetris, where 50% prefers the balanced version, 25% prefers the trivial version and 25% prefers the normal version. Finally, only one volunteer was considered an expert and his favourite version was the fit version.

Another interesting analysis is to know which version of Tetris generates the highest flow state. Therefore, we filtered the flow questionnaires by version type, and as a result, we had three groups, each with 40 completed questionnaires. In each group, we calculated the average flow state experienced by the players. We compute each questionnaire nine dimensions average and then, we calculate the result's average. Volunteers had an average flow status of $3,778$ when playing the normal version, had an average flow status of 4 when playing the balanced version, and had an average flow status of $4,0028$ when playing the trivial version (table III). Furthermore, we compared the average flow state between the different Tetris versions. We found that the average flow state of the balanced version is significantly higher (M=4, SE=0.602) than the average flow state of the normal Tetris version (M=$3,778$, SE=0.54, p$<$ .01). Therefore, we accept *H1, we get more players to experience flow using DDA*. In the S-FSS, each question represents a flow dimension and items 1, 2, 6 had a significant difference when comparing normal games against adjusted ones (table III). Hence, adjusted games have a better challenge-skill balance (item 1), players experienced a better fusion of action-awareness (item 2), and a greater sense of control (item 6).

Then, we grouped beginner, intermediate and advanced players using the total score they received in the normal games to accept or reject *H2, it is harder for beginners to experience flow compared to intermediate or advanced level players*. The group of beginners contains 21 normal games, where the minimum total score is 640, the maximum total

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

Flow Dimension	**Average Flow**			**p-value**	
	normal	**dda**	**easy**	**normal vs dda**	**dda vs easy**
Challenge-Skill Balance	3,9	4,3	4,3	**0,033**	0,881
Merging of Action and Awareness	3,4	3,8	3,8	**0,003**	0,756
Clear Goals	4,0	4,2	4,4	0,173	0,291
Unambiguous Feedback	4,3	4,3	4,5	0,596	0,281
Concentration on the Task at Hand	4,4	4,5	4,3	0,512	0,405
Sense of Control	3,3	3,7	3,9	**0,045**	0,369
Loss of Self-Consciousness	4,2	4,3	4,5	0,256	0,221
Transformation of Time	3,5	3,6	3,7	0,376	0,711
Autotelic Experience	3,1	3,4	2,8	0,212	**0,007**

score is 2725, and the average score is 1601. The intermediate players' group had 19 players, with a minimum total score of 3075, a maximum total score of 4720, and an average total score of 3969. There were no advanced players in this experiment based on the total score obtained in the normal games. We compare the average flow score between beginners and intermediate players in each group, i.e. beginner's normal games vs intermediate's normal games, beginner's balanced games vs intermediate's balanced games, and beginner's easy games vs intermediate easy games. If intermediate players have a significantly higher average flow score than beginners, then the hypothesis is accepted. Data shows no significant difference in the average flow score between beginners and intermediate in any group (tables IV and V). Therefore, we reject *H2*. However, there are differences when comparing normal and balanced games in the same player's profile group. The balanced games from beginners are significantly higher in challenge-skill balance and merging of action-awareness than normal ones (tables IV). And balanced games from intermediate players are significantly higher in merging of action-awareness, sense of control and autotelic experience than normal ones (table V).

Additionally, we compared the trivial version with the balanced version and found no significant difference between the flow state averages. However, when comparing the averages by the dimension of flow (table III), we have that the average score of item 9 from the balanced version group is significantly higher (M=3.4, SE=1.05) than the average of item 9 from the trivial version group (M=2.8, SE = 1.32, $p < .01$). Item 9 tries to identify the autotelic experience a person senses when performing a task. In the next section, we explain in detail what this means.

Last, we compare the average flow score from the trivial and normal games to either accept or reject *H3, if we provide an easy Tetris version non-expert players, then they will experience flow*. For both trivial and balanced versions, the game is easier than normal Tetris. The balanced version adapts the challenges to be easier when necessary. The trivial version always provides help. The results show that the average flow state of the trivial version significantly higher (M=4, SE=0.55) than the average flow state of the normal version (M=$3,778$, SE=0.54, $p < .05$). These results show that both versions that provide easier challenges to the players helped them experiencing flow stronger than a normal game. Therefore, we accept *H3*. Plus, we reject *H4, if Tetris is too trivial, then the players won't experience flow*, because even though the trivial version mostly returns squares or I, players entered flow in our experiment.

VII. 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.
- Players have a greater sense of challenge-skill balance, fusion of action-awareness, and sense of control when playing the balanced version compared to the normal version.
- The average flow experience from the balanced version is not significantly higher than the trivial version.
- The balanced version produces a greater autotelic experience than the trivial version.
- It is not harder for beginners getting into flow compared to intermediate and advanced players.

We expected players to find the trivial version boring compared to the balanced version, but this was not the case. And we found a few possible reasons why. The first one is that most players consider themselves beginners, and based on the flow channel, the lower the skills, the lower the challenge should be. Then, a player with few skills is more likely to enter flow when the required skills are low (adequate for his low skill level). Another reason is that players score better on the trivial version (average total score is 4720) when comparing the balanced one (average total score is 3659) because the game delivers easy-to-place pieces all the time, such as squares or I. And having a better total score can influence the answers to the questionnaire. In addition, a Tetris Analytics game lasts around 3-5 minutes, and during this time, the player cannot get bored enough for having such a simple game. But it is likely that if we ask a person to play the same trivial game for a longer period, then they will eventually find it tedious and rate it worse than if they only played a few minutes. Also, the autotelic experience is significantly higher in the balanced version than in the trivial version when analysing the averages per flow dimension. The flow theory has its origins in the desire to understand the autotelic personality phenomenon, where a person tends to enjoy life or things for their own good and not because they are going to obtain some external goal [11]. The autotelic experience describes

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

Beginners						
Flow Dimensions	**Average Flow**			**p-value**		
	normal	**dda**	**easy**	**normal vs dda**	**easy vs dda**	**normal vs easy**
Challenge-Skill Balance	3,810	4,333	4,143	**0,038**	0,446	0,358
Merging of Action and Awareness	3,476	3,810	3,714	**0,031**	0,705	0,397
Clear Goals	3,810	4,048	4,333	0,382	0,343	0,126
Unambiguous Feedback	4,238	4,286	4,476	0,847	0,329	0,348
Concentration on the Task at Hand	4,429	4,429	4,429	1,000	1,000	1,000
Sense of Control	3,333	3,571	3,714	0,382	0,666	0,337
Loss of Self-Consciousness	4,143	4,333	4,524	0,258	0,446	0,072
Transformation of Time	3,619	3,762	3,810	0,379	0,789	0,407
Autotelic Experience	3,238	3,238	3,190	1,000	0,847	0,895

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

Intermediate Players						
Flow Dimensions	**Average Flow**			**p-value**		
	normal	**dda**	**easy**	**normal vs dda**	**easy vs dda**	**normal vs easy**
Challenge-Skill Balance	4,0000	4,1579	4,4211	0,4535	0,2350	0,0880
Merging of Action and Awareness	3,3158	3,7895	3,7895	**0,0462**	1,0000	**0,0349**
Clear Goals	4,2105	4,4211	4,4737	0,2590	0,6669	0,0562
Unambiguous Feedback	4,2632	4,3684	4,4211	0,4291	0,6669	0,2680
Concentration on the Task at Hand	4,2632	4,4737	4,2105	0,4291	0,3306	0,8413
Sense of Control	3,3158	3,8421	4,0526	**0,0465**	0,2973	**0,0017**
Loss of Self-Consciousness	4,1579	4,2105	4,4211	0,7162	0,3306	0,0562
Transformation of Time	3,3684	3,4737	3,5263	0,6669	0,8041	0,5778
Autotelic Experience	3,0000	3,4737	2,3684	**0,0245**	**0,0004**	0,0619

the intrinsically rewarding experience an individual feels when on flow [1]. Csikszentmihalyi [10] describes this dimension as the result of the other eight flow dimensions. Moreover, flow is considered such an enjoyable experience that, once experienced, it becomes a desired state. So it is considered by many as the ultimate motivation that drives them to continue pushing towards higher limits. Although the balanced version does not have a significantly higher flow average when comparing to the trivial version, the former provides a more rewarding experience. Consequently, the players enjoy more the balanced version than the trivial one.

VIII. CONCLUSION

This work presents a methodology that serves as a guide to implementing a DDA system in video games. In this methodology, we use flow theory to adjust the difficulty of the game. Plus, Tetris Analytics for our experiments. The data show that the average flow experience when playing a balanced version is significantly higher than the normal version. In this case, players have a greater sense of challenge-skill balance, merging of action-awareness, and sense of control when playing the balanced version compared to the normal version. We also found no overall significant difference between the flow experienced between trivial and balanced versions. However, we found a significant difference in the autotelic experience dimension when considered each separately.

To better understand the implications of our results, we could include biometrics to measure flow experience in specific parts of the game and compare it to the control measurement and the adjustment made. This way, we could evaluate the effectiveness of an adjustment by examining the player's biometric after the adjustment. Although the results of this experiment were positive, they also raise new questions. We found no significant differences in average flow experience when we compared the balanced and trivial versions. However, we did find a significant difference in the autotelic experience dimension. A lengthy experiment focused on novice players could help us identify the differences between the appropriate level of difficulty for this specific group of players.

ACKNOWLEDGMENT

The authors would like to thank...

REFERENCES

[1] 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.

[2] 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.

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

[4] 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.

[5] 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.

[6] 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.

[7] M. Jennings-Teats, G. Smith, and N. Wardrip-Fruin, "Polymorph: dynamic difficulty adjustment through level generation," in *Proceedings of the 2010 Workshop on Procedural Content Generation in Games*. ACM, 2010, p. 11.

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

[9] 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.

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

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

[12] R. Sutoyo, D. Winata, K. Oliviani, and D. M. Supriyadi, "Dynamic difficulty adjustment in tower defence," *Procedia Computer Science*, vol. 59, pp. 435–444, 2015.

[13] A. Denisova and P. Cairns, "Adaptation in digital games: The effect of challenge adjustment on player performance and experience," in *Proceedings of the 2015 Annual Symposium on Computer-Human Interaction in Play*. ACM, 2015, pp. 97–101.

[14] A. Hintze, R. S. Olson, and J. Lehman, "Orthogonally evolved ai to improve difficulty adjustment in video games," in *European Conference on the Applications of Evolutionary Computation*. Springer, 2016, pp. 525–540.

[15] M. Fagan and P. Cunningham, "Case-Based Plan Recognition in Computer Games," in *Case-Based Reasoning Research and Development*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 161–170.

[16] D. Charles and M. Black, "Dynamic Player Modelling: A Framework for Player-centred Digital Games," *Proceedings of 5th International Conference on Computer Games: Artificial Intelligence, Design and Education (CGAIDE'04)*, pp. 29–35, 2004.

[17] 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.

[18] O. Missura and T. Gärtner, "Player Modeling for Intelligent Difficulty Adjustment," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5808 LNAI, pp. 197–211, 2009.

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

[20] R. Holt and J. Mitterer, "Examining video game immersion as a flow state," *108th Annual Psychological Association, Washington, DC*, 2000.

[21] L. Michailidis, E. Balaguer-Ballester, and X. He, "Flow and immersion in video games: The aftermath of a conceptual challenge," *Frontiers in Psychology*, vol. 9, p. 1682, 2018.

[22] P. Cairns, A. Cox, and A. I. Nordin, "Immersion in digital games: review of gaming experience research," *Handbook of digital games*, vol. 1, p. 767, 2014.

[23] M. Csikszentmihalyi and I. S. Csikszentmihalyi, *Optimal experience: Psychological studies of flow in consciousness*. Cambridge university press, 1992.

[24] R. W. Quinn, "Flow in knowledge work: High performance experience in the design of national security technology," *Administrative science quarterly*, vol. 50, no. 4, pp. 610–641, 2005.

[25] J. Heo, Y. Lee, P. M. Pedersen, and B. P. McCormick, "Flow experience in the daily lives of older adults: An analysis of the interaction between flow, individual differences, serious leisure, location, and social context," *Canadian Journal on Aging/La Revue canadienne du vieillissement*, vol. 29, no. 3, pp. 411–423, 2010.

[26] 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.

[27] M. G. Jones, "Learning to play, playing to learn: Lessons learned from computer games," *Association for Educational Communications and Technology*, 1997.

[28] ——, "Creating electronic learning environments: Games, flow, and the user interface." *ERIC*, 1998.

[29] B. Cowley, D. Charles, M. Black, and R. Hickey, "Toward an understanding of flow in video games," *Computers in Entertainment (CIE)*, vol. 6, no. 2, pp. 1–27, 2008.

[30] D. Pavlas, "A model of flow and play in game-based learning the impact of game characteristics, player traits, and player states," Ph.D. dissertation, University of Central Florida, 2010.

[31] D. Lora, A. A. Sánchez-Ruiz, P. A. González-Calero, and M. A. Gómez-Martín, "Dynamic difficulty adjustment in tetris," in *The Twenty-Ninth International Flairs Conference*, 2016.

[32] 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.

[33] H. Romdhane and L. Lamontagne, "Reinforcement of local pattern cases for playing tetris." in *FLAIRS Conference*, 2008, pp. 263–268.