Flow-Z: A Flow-based Adaptable Game to Maintain Optimal Challenge

Tomás Alves
Department of Computer Engineering
Instituto Superior Técnico
Lisboa, Portugal
tomas.alves@tecnico.ulisboa.pt

Abstract—Often video games fail to attract a wider range of consumers because people become uninvolved when they fail to meet the in-game difficulty. This dissertation addresses the problem of the in-game difficulty not being correctly adapted to the gamers, leading to their uninterest for not having their skills balanced with the challenge of the game. This balance is one of the conditions that lead people to flow, which is the mental state associated with optimal enjoyment of an activity. In our work, we study if flow may be relevant for gameplay adaptability and may offer a better gaming experience, since it provides a better enjoyment of an activity. We created a hypothesis to verify if a game that adapts its parameters to a representation of the mental state of the player following the flow theory can provide a better gaming experience compared to a game that adapts to their performance. We developed a first-person shooter video game that adapts its in-game difficulty and environmental settings based on a representation of their mental state to keep a balance between the skills of the player and the challenge of the game. The mental state of the player is measured with their physiological signals, namely the heart rate and the beta band of the brainwaves, and we distinguish the mental state of the player with an accuracy of 87%. We also conducted an evaluation using self-perceived flow and in-game scores as metrics to compare the mental state-based adaptability with a performance-based version. Results show that the latter provided a better gaming experience.

Index Terms—Video games, flow, performance, adaptable gameplay, psychophysiology

I. INTRODUCTION

Since traditional game design offers a discrete scope of difficulty, players may end up feeling alienated for not falling on the discriminatory set of in-game difficulties, i.e. some players may find a game that is either too easy or too hard for them. This dissertation addresses the problem of the in-game difficulty not being correctly adapted to the gamers, leading to their uninterest for not having their skills paired with the challenge of the game.

The balance between the challenge of an activity and the skills of an individual is one of the conditions that lead people to flow. This concept is associated to the mental state that people feel when they are completely engaged in an activity and have an optimal experience while performing it [1]. A game that adapts its gameplay difficulty and environment interactions allows a better gaming experience, using the former to balance the challenge of the game and the skills of the player and the latter to create a greater sense of engagement in the player. Therefore, we investigate if flow may be relevant for gameplay adaptability and may offer an enhanced gaming experience, since flow provides a better enjoyment of an experience.

Our hypothesis is that gamers have a better gaming experience playing a game that adapts to the representation of their mental state by keeping them in flow compared to a game that adapts to their performance. We created and ran two prototypes in user testing to evaluate self-perceived flow and in-game performance: one adapts to the mental state of the user and the other to their performance. As a metric for success, we defined that players have a higher performance and a higher self-perceived flow when they are playing a game that adapts to their mental state compared to playing a game which gameplay adapts to their performance.

This paper is organized as follows: we start by presenting the concept of flow, taking into account its physiology; afterwards we present a system overview will all the developed work and describe it; we continue by presenting and discussing the results of the user tests; finally, we finish with our conclusions and raise a direction for future investigation.

II. BACKGROUND

Flow regarding human behavior and computers has most notably been studied in user experience and as a means to explain user engagement. This section presents the flow theory and how we can physically measure the flow state.

II-A. Flow

Csikszentmihalyi [2] addresses the feeling of deep engagement as state of "flow". Based on his findings, Csikszentmihalyi defined the two conditions for flow: (1) Perceived challenges of the activity match and stretch the capabilities of the individual, thus producing an experience of being fully engaged in the task and acting on the height of their skills [3], [4]; and (2) The goals of the activity are explicit and reachable, and one receives instant feedback for their progress on the activity [4].

Nacke and Lindley [5] presented a two-dimensional four-channel model of flow based on Csikszentmihalyi [1] and Ellis et al. [6] which incorporates the apathy state and is used most
frequently for describing games and gameplay experience. This model is illustrated in Figure 1.

![Four-channel flow model](image)

**Fig. 1:** Four-channel flow model presented by Nacke and Lindley [5]. The flow state (an equilibrium between skills and challenge) is compared against anxiety (challenge exceeds skills) and boredom (skills exceed challenge) conditions. Apathy was reported when challenges and skills were too low at the start or when a task had to be repeated frequently.

### II-B. Flow in Experimental Settings

The first-level physiological indicators that are proven to be effective measuring flow are the Heart Rate (HR) and Heart Rate Variability (HRV) [7]–[9] for cardiovascular activity, the Respiratory Rate (RR) [7], [9] for respiratory activity and the alpha, low beta and theta bands [10], [11] for the Electroencephalography (EEG). The decrease of the Galvanic Skin Response (GSR) is also an indicator for higher arousal and can be effectively used to measure it [12], [13].

Due to technical limitations, this study only uses HR, GSR and EEG to assess the player’s mental state. These signals were also selected because they are rather resistant to movement artifacts and can be measured non-invasively.

### III. INITIAL DEVELOPMENT

In order to create a game which gameplay adapts to the user’s mental state, we need three different components. First, the game itself which must implement a gameplay adaptation mechanism. With this mechanism, it is possible to change the game parameters while the user is playing the game. Thus, we can change the challenge of the game by shifting the game parameters. Second, it is necessary to know the mental state of the player so that we can take a proper adaptation. We need a classification framework which function is to read the physiological signals of the player and label the output as the current mental state of the player. Third, by knowing the current mental state of the player and taking into account the Flow Theory, we can decide on what to change in the game parameters so that the player is in a flow state. The controller framework is responsible for reading the current mental state of the player and varying the game parameters, depending on the previous ones. Figure 2 illustrates the relation between these three components and how the user is present on that relation.

![User and the three core components](image)

**Fig. 2:** User and the three core components of the solution. From top to bottom, the User, the Classification Framework, the Controller Framework and the Game. While the User is playing the Game, the Classification Framework is reading their biofeedback and processing it. The result is the current mental state of the player. The Controller Framework reads the current mental state of the player and decides the next game level. The Game changes its parameters accordingly to the current game level.

### III-A. Game

The game is a First-Person Shooter (FPS) in a dark environment with cartoonish zombie figures as enemies, based on the Unity3D "Survival Shooter tutorial"\(^1\), as illustrated in Figure 3.

The choice of a first-person camera is that it will easily lead to user immersion [14], given that one could see immersion as a precondition for flow. Note that immersion involves a

\(^1\)www.unity3d.com/learn/tutorials/projects/survival-shooter-tutorial
loss of a sense of context, while flow describes a level of complete involvement \cite{3}. This is possible given the removal of the avatar representation and putting the player in first-person perspective, which leads the player to feel like they are acting directly upon the virtual game world. \cite{15} Thus, the player can fully identify with the game character represented only by the weapons that reach into the game environment. Takatalo et al. \cite{16} and Nacke and Lindley \cite{5} also used an FPS game to study user engagement.

The zombie genre gained popularity in recent years due to the development of themes like \textit{The Walking Dead}\footnote{\url{www.thewalkingdead.com}}, \textit{Resident Evil}\footnote{\url{www.en.wikipedia.org/wiki/Resident_Evil}} and \textit{Left 4 Dead}\footnote{\url{www.en.wikipedia.org/wiki/Left_4_Dead}}. Being an endless shooter game, the product of this work uses the challenge to survive owned by the zombie genre and familiarity with the theme to create user engagement.

This section covers the base design and the development of mental state inducement versions.

\subsection*{III-A1 Base Game}

Figures 4, 5 and 6 depict the three different enemies used in the tutorial: Zombunnies (low health points and high speed), Zombears (medium health points and medium speed) and Hellephants (high health points and low speed).

We introduced several new features including the first-person camera, a rocket launcher, stamina and sprinting system and a pick-up system with rewards of health and ammunition that the player can catch scattered across the game world. These features were chosen based on the work of Cowley \textit{et al.} \cite{17}, since they mapped flow and gameplay elements according to Csikszentmihalyi’s dimensions of flow \cite{3}.

\subsection*{III-A2 Inducement Versions}

A key point in our work is that our contribution must be able to distinguish the mental state of the user. Therefore, we need to have data of the different mental states. We created four different versions thereof by varying some of the game parameters - the speed, health and spawn time of the enemies. The different versions were designed taking into account existing studies about flow during gameplay \cite{5}, \cite{7}, as well as informal feedback from users. Figures 7, 8 and 9 show sketches based on player’s intuition regarding how users felt while play testing differences in the game parameters values of enemies’ speed, health and spawn time stats. The plots are not validated.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{Figures/Fig_7.png}
\caption{Sketch of the mental states induced by the game as a function of the character parameters health and speed. If all characters are too slow and easy to kill, the game is too easy and the player becomes bored. As the value of the two parameters increases, the user becomes increasingly engaged and eventually anxious and frustrated. Finally, if the characters are too slow but have excess health, they are easy to kill but doing so takes too much time, eventually inducing apathy in the player.}
\end{figure}

Each version was fine-tuned to induce in the player a specific mental state, such as anxiety, boredom, engagement and frustration. The choice of which mental states to induce was based on the works of Csikszentmihalyi \cite{1}, Gilleade and Dix \cite{18}, and Poels \textit{et al.} \cite{19}. Players will also be able to play a sandbox version of the game and develop their initial skills. This takes away the need to create a game version focusing on apathy, since participants will not have skills too low for the task.

The versions and their variations are presented in Table I. In addition to the game parameters values of enemies’ speed, health and spawn time stats, we decided to alter other variables in versions B and D. In version B, the player’s speed is
Fig. 8: Sketch of the mental states induced by the game as a function of the enemy parameters speed and spawn time. Testers said that, independently of the speed, for high values of spawn time, they would be bored and, in the limit, apathetic. As the value of the speed parameters increases and the spawn time decreases, the user becomes decreasingly engaged and eventually anxious and frustrated.

Fig. 9: Sketch of the mental states induced by the game as a function of the enemy parameters health and spawn time. A balance between the values of the two parameters leads the user to engagement. If enemies have a short spawn time, the player becomes anxious, but, if enemies also have excess health, the player becomes frustrated. Finally, if the spawn time is high, players become bored, except if they also have excess health, which leads players to an apathetic state.

In order to validate the inducement versions, we performed a user testing phase. In this testing phase, we also collected biofeedback in case we could validate our versions to use it to model the classification framework. Regarding the guidelines we used for the inducement versions, results suggest that versions A, B and C induce anxiety, boredom and engagement, respectively. Moreover, version D significantly induces frustration. Therefore, we can use physiological data collected in this testing phase to model our classifier.

### Table I: Version name, the mental state it is intended to induce and the general characteristics of each game version.

<table>
<thead>
<tr>
<th>Version</th>
<th>Mental State</th>
<th>Parameters Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Anxiety</td>
<td>Low enemy health</td>
</tr>
<tr>
<td></td>
<td></td>
<td>High enemy speed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Low enemy spawn time</td>
</tr>
<tr>
<td>B</td>
<td>Boredom</td>
<td>High enemy health</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Low enemy speed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>High enemy spawn time</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Low player speed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>High pick-up spawn time</td>
</tr>
<tr>
<td></td>
<td></td>
<td>High ammunition pick-up probability</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Low health pick-up probability</td>
</tr>
<tr>
<td>C</td>
<td>Engagement</td>
<td>Medium enemy health</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Medium enemy speed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Medium enemy spawn time</td>
</tr>
<tr>
<td>D</td>
<td>Frustration</td>
<td>Low enemy health</td>
</tr>
<tr>
<td></td>
<td></td>
<td>High enemy speed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Low enemy spawn time</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Low shooting probability</td>
</tr>
</tbody>
</table>

In order to validate the inducement versions, we performed a user testing phase. In this testing phase, we also collected biofeedback in case we could validate our versions to use it to model the classification framework. Regarding the guidelines we used for the inducement versions, results suggest that versions A, B and C induce anxiety, boredom and engagement, respectively. Moreover, version D significantly induces frustration. Therefore, we can use physiological data collected in this testing phase to model our classifier.

### III-B. Classification Framework

The main function of the classification framework is to detect the mental state of the user. The detection of the mental state is carried out by a classifier. As mentioned in the previous section, all data obtained from performing the previous tests to the users is the feedstock for the classifier. We need to model and train the classifier with these data so that the final solution can distinguish the mental states we aim to address among themselves. This section addresses the validation of the physiological measures to be used when distinguishing the mental state of the user and the modeling of the classifier. We also present the final architecture of this main component.

### III-B1. Choosing Physiological Measures

We collected physiological data from thirty users with a Plugged Kit. Before playing the inducement versions, we recorded data for the baseline values and players had access to

www.bitalino.com/en/plugged-kit
to the sandbox. Each participant played the different versions in a random order for ten minutes each.

Before analyzing data, we processed the physiological measures. Bitalino reads the analog signals from the participants and transforms them into digital signals. These signals are not meaningful per se and we have to convert them to the physiological measures presented in section II-B. Thus, we converted them to Electrocardiography (ECG), Electrodermal Activity (EDA) and EEG values. After that we used the BioSPPy toolbox\(^6\) to compute the instantaneous HR and GSR, and EEG alpha, beta and theta bands.

Concerning the results, alpha and theta bands were not significant in the distinction between the version the player was playing. Yet, the beta band allows to distinguish version C from B and D. We were not able to have a brainwave band that could significantly distinguish between all combinations of two versions. Alpha and theta bands did not have values that were significantly different between versions, yet the beta band proved to have significant differences. We can conclude that this is the best band of brainwaves to use when we are trying to distinguish the mental state of the player.

Analysis for the HR and GSR also did not provide any other physical measure that could effectively differentiate between the mental states. We discarded the usage of GSR to detect variations in the mental state because the appearance of missing values was too common to provide a reliable analysis.

Since we did not aim to create a game which could only adapt to the player based on one physiological feature, we decided to further analyze the values of the alpha and theta bands and the HR. We decided to follow-up with Wilcoxon tests to see how significant the differences were in these three signals between the versions. After analyzing the results, we concluded that the measure that better complements the beta band is the HR.

### III-B3 Choosing a Classifier

With our features defined, we can start testing several classifiers. We chose to test four classification algorithms: Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM) and Multilayer Perceptron (MLP). Firstly, we randomly changed the order of the lines from the data set so that we can have an even label distribution among the whole set. After that, we took 20% of the data to use as the test set for all the classification algorithms. Then, we normalized data by mean and variance. We used the scikit-learn toolkit\(^7\) to process data and model the classifiers. We modeled each of the four classifiers using a 10-fold cross-validation grid search method with a set of hyper-parameters. This fine tuning technique lets us increase, in our case, the accuracy with hyper-parameters that change the behavior of the classifier. We chose hyper-parameters ranges with values inside the usually used ranges and not in the extremes. This way, we obtain the classifiers with the highest accuracy and its parameters.

After modeling the four different classifiers, we have to choose which one we use in our game. Tables II and III show all the relevant values in which we based our choice.

#### TABLE II: Comparison of the classifiers with average test set results for accuracy, precision, recall and f1-score.

<table>
<thead>
<tr>
<th></th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>DT</td>
<td>0.72</td>
<td>0.75</td>
<td>0.75</td>
<td>0.75</td>
</tr>
<tr>
<td>RF</td>
<td>0.85</td>
<td>0.86</td>
<td>0.85</td>
<td>0.85</td>
</tr>
<tr>
<td>SVM</td>
<td>0.83</td>
<td>0.83</td>
<td>0.83</td>
<td>0.83</td>
</tr>
<tr>
<td>MLP</td>
<td>0.87</td>
<td>0.87</td>
<td>0.87</td>
<td>0.87</td>
</tr>
</tbody>
</table>

#### TABLE III: Comparison of the classifiers with average test set results for support to each condition.

<table>
<thead>
<tr>
<th></th>
<th>Sup. Anxiety</th>
<th>Sup. Flow</th>
<th>Sup. Boredom</th>
</tr>
</thead>
<tbody>
<tr>
<td>DT</td>
<td>298</td>
<td>294</td>
<td>54</td>
</tr>
<tr>
<td>RF</td>
<td>336</td>
<td>285</td>
<td>25</td>
</tr>
<tr>
<td>SVM</td>
<td>302</td>
<td>304</td>
<td>40</td>
</tr>
<tr>
<td>MLP</td>
<td>310</td>
<td>292</td>
<td>44</td>
</tr>
</tbody>
</table>

The MLP showed higher accuracy, precision, recall and f1-score compared to the other three classifiers. Regarding support, RF has the higher value for anxiety, SVM for flow and DT for boredom. Although MLP does not have a higher support for the mental states, no other classifier dominates these three last measures as the MLP dominates the first four.

\(^6\)www.github.com/PIA-Group/BioSPPy

\(^7\)http://scikit-learn.org/stable/
This leads us to conclude that the best classifier for our work is the MLP.

### III-B4 Architecture

Figure 10 depicts the final architecture of the classification framework. The illustrated process occurs for time intervals of twenty seconds. At each twenty seconds, a bash script runs the BioSPPy toolkit and processes the text file from OpenSignals (r)evolution. BioSPPy outputs two text files: one has the instantaneous HR and the other the values for the beta band. Next, the bash script calls a Python script which extracts the features from those files, loads the classifier and writes on a file the classified mental state.

![Diagram of the classification framework](image)

#### III-C. Controller Framework

The third and final main contribution is the controller framework. It is responsible for changing the parameters of the game based on the mental state of the player. In order to create a system that has to keep information regarding the current state, a list of possible states to transit to and a matrix of transitions, we opted for a state machine. Therefore, we implemented a state machine in our adaptable game which allows the game engine to know the current state and, when it receives the up-to-date mental state of the user, it transits to the next state depending on it. This section presents how the state machine was developed and how the game conducts the adaptation of the game parameters and environment settings.

#### III-C1 State Machine

We opted to create a game that has three different levels. The mental state of the player is the trigger that allows the gamer to travel between the different levels while they are playing. The transitions the state machine (see Figure 11) receives as input are the output of the classification algorithm. After the state machine updates the current state, the game adapts itself to it. It is similar to the one developed by Rani et al. [20]. We did not find that more states would give a better experience, since three were enough to represent changes in the gameplay.

We defined that level 1 would be the state with the least engagement elements and the easiest in terms of difficulty. On the contrary, level 3 would have all the engagement elements and it would be the hardest. This way we can always keep the player in a flow state. As an example, if the difficulty is too high for their skills, they will become anxious; the state machine must decrease the game difficulty which is equivalent to go down one level. With a lower difficulty, players will reach a flow state. While their skills match the difficulty of that level, they will be in flow and developing their skills until they are too high for the challenge of that level. In that case, the player will be bored and the state machine must go one level up. With an increase of difficulty, they will reach flow again and continue on developing their skills. This process is the core of the game that allows players to have the most enjoyable experience while they are playing the game because it tends to always leave them in a flow state. As we mentioned in Section I, we would compare this state machine against one that would adapt to the player’s performance. We created a second version of the game with a state machine that adapts to the performance.

With the number of levels defined, we needed to create the guidelines for the adaptation. The design criteria to keep the user in a flow state are concentrated on two different fronts: environmental settings to adapt engagement and enemies stats alteration to adapt the difficulty.

#### III-C2 Environmental Settings

As mentioned before, we needed to create elements in the game that would lead the player to a higher engagement and, consequently, to flow. The design guidelines for the environmental settings are the following: (1) in levels 2 and 3, when enemies die they explode giving Area of Effect (AOE) damage to nearby enemies, (2) in level 3, barrels
Fig. 11: Three states machine diagram for mental state-based gameplay adaptation. The player’s mental state is classified as one of the following states: Anxiety, Boredom and Flow. At each specific time interval, the game checks which mental state the classifier chose and feeds it to the state machine. After getting that transition, the state machine processes it and updates the state. When the game starts, the player starts at level 1.

spawn randomly on the environment and give AOE damage to nearby enemies when they explode, (3) color of explosions depends on the flow of the player, and (4) fitting sensory effects (explosions, sounds, textures).

With these guidelines, we change the way the player interacts with the enemies thereby creating engagement. The objective is not to prolong the task of killing an endless wave of zombies, since that can lead to boredom. This way, the player can create new strategies based on the level. The differences in the color of explosions is a passive way of showing the player that the level changed.

III-C3 Difficulty Settings

Regarding the difficulty adaptation, as higher the level, the difficulty is also harder. We decided that enemies gain speed, health and a shorter spawn rate as the player goes up in levels, gradually increasing combat difficulty. The guidelines presented in the previous section allow the player to fight harder enemies, since killing one or shooting at a barrel allows the player to deal damage to any number of surrounding enemies. This balance is responsible for creating a higher engagement although the enemies are harder to kill.

IV. Validation

This section introduces the second testing phase of this work, presents the results and discusses them.

IV-A. Method

In this section we describe the population sample used in our study, material we used in the tests and the procedure used in our second testing phase.

IV-A1 Participants

We recruited anyone interested in participating with at least 18 years old through standard procedures, including direct contact and through word of mouth. Again, participants were asked to sign a consent form and the experience presented no potential risks and no anticipated benefits to them.

We conducted a total of 21 tests. All tests occurred between 08:00h and 20:00h. There were no technical problems this time, so we kept a total of 21 completed tests for the analysis.

The participants (16 males, 5 females) were ranged in age from 19 to 27 ($M = 22.43, SD = 1.91$). From all the participants, eleven had already been present in the first testing phase.

Only one participant reported no video game-playing time. The other participants play at least once a day (28.57%), at least once week (42.86%) or at least once a month (23.81%). Fifteen participants frequently play FPS. Again, a mixed Analysis of Variance (ANOVA) showed that playing in our first testing phase, gaming periodicity and gender had no significant effects on results.

IV-A2 User Evaluation

This testing phase had a consent form and an adapted version of the Game Engagement Questionnaire (GEQ) [21]. We used five items from the adapted GEQ. Four address flow and one presence. We use them to see which version led to a higher self-perceived flow.

IV-A3 Apparatus

We needed two computers for this testing phase. Data was recorded at 100 Hz with OpenSignals (r)evolution Mac OS X (v.2017)$^8$ software. Both computers were capable of processing and running the programs without any delay. Players interacted with the game through mousepad and headphones. We also carefully chose a set of images, most of them from the International Affective Picture System (IAPS) [22] to lead to a relaxing state, as done in [23], [24]. One of the images is depicted in Figure 12.

Fig. 12: Pictures with flowers and gardens were proved to lead to high values of valence and low values of arousal, which correspond to relaxing stimuli. This photo has the identifier "Garden - Picture 5202" in the IAPS.

We used a BITalino$^9$ to record the Blood Volume Pulse (BVP) with a photoplethysmography sensor and the electricity on the forehead with three electrodes. The first allows us to know the instantaneous HR and the second the brainwaves (alpha, beta and theta bands). All these items can be obtained in the (r)evolution Plugged Kit BT$^{10}$ with an addition of a PulseSensor$^{11}$. The BITalino was placed on a table behind the

$^8$www.bitalino.com/en/software
$^9$www.bitalino.com/en/hardware
$^{10}$bitalino.com/en/plugged-kit-bt
$^{11}$store.plux.info/bitalino-sensors/42-pulseSensor.html
Users and the sensors connected to it in each experimental condition and during the baseline measurement. While they were playing, participants had an earplug attached to one of their ears and two electrodes on their forehead (positions FP1 and FP2 in the 10–20 system [25]), one electrode on the left side of their neck (its function is to serve as "ground" for the difference in both hemispheres). The second band was only used for participants who had long hair covering their forehead. Users were seated on a chair with the computer in front of them on top of a table. Other materials used for the experiences were a headband, neurodiagnostic electrode paste, alcohol and cloth.

**IV-A4 Procedure**

Tests were performed in room 0.09 of Pavilhão de Informática II in campus Alameda of Instituto Superior Técnico. Temperature was kept between 21°C and 23°C in order to prevent the electrode gel from melting and displacing the sensors.

The assistant started by explaining to the users the purpose of the study, what they would be doing and that they should not move their head while they were playing in order to prevent detaching the electrodes. After it, we asked the users to fill a consent form and a form regarding their demographic data and gaming experience. Then, we placed the physiological sensors on the user. We recorded the baseline values for five minutes with our set of images.

After that, users played a sandbox version of the game to try the sensitivity and in-game interactions. The assistant lets users play as much time as they wanted so that they could develop the minimum skills to play the game in the different versions. Even users that had participated in the first testing phase had the minimum skills to play the game in the different versions. The assistant started the game for the tester.

1) The assistant verified if every sensor was correctly placed;
2) The assistant asked the tester if they was comfortable and ready to play;
3) The assistant started the game for the tester;
4) If the player died, the game automatically restarted;
5) The player played for ten minutes and after it the game automatically restarted;
6) The assistant asked the tester to fill the form addressing the version the latter played;
7) The player rested for three minutes looking at a picture in order to return to a neutral mental state; and
8) After three minutes, the assistant repeated this procedure to the next version, if there was any other version to play.

The order of the adapted GEQ items was randomized between the two versions. We kept the gaming closing itself so that we would not break the participant’s immersion in the game. After the tester played the two versions, the assistant removed the sensors from the tester and cleaned them with alcohol to remove the electrode gel. Free comments were also invited. Testers received a compensation based in candies. We created a contest to see which player achieved the highest score across all gaming sessions. The winner received a gift card worth 20EUR.

**IV-B. Results**

One of the objectives of this work is to prove that the self-perceived flow is higher when the user is playing a game that adapts to their mental state compared to playing a game which gameplay adapts to their performance. We use data obtained from the Adapted GEQ to perform those comparisons. We can then create a hypothesis:

**H1:** Users have a higher self-perceived flow playing the mental state-based version compared to the performance-based

We conducted a paired-samples t-test to compare the self-perceived flow using the normalized Flow Degree values between the mental state-based and the performance-based versions. There was a significant difference in the self-perceived flow for the performance-based ($M = 6.46, SD = 1.29$) and mental state-based ($M = 6.05, SD = 1.34$) conditions; $t(20) = 2.215, p = 0.039$. These results show that the version the participant played has an effect on their self-perceived flow; more specifically that the performance-based prototype leads to a greater self-perceived flow compared to the mental state-based one. Therefore, we refute our hypothesis.

Other objective was to prove that a player has a higher performance when the user is playing a game that adapts to their mental state compared to playing a game which gameplay adapts to their performance. We decided to compare this performance by the sum of all scores players would have by the end of each time they played. When an enemy was killed it provided a certain amount of points to the player. If it was a Zumbunny or a Zombrear, the player would win 10 points; and 50 points if it was an Hellephant. We summed all the scores each player obtain while playing each version and created the last hypothesis:

**H2:** Users have higher scores playing the mental state-based version compared to the performance-based

A paired-samples t-test was conducted to compare the scores in the mental state-based and the performance-based versions. There was a significant difference in the scores for the performance-based ($M = 4820, SD = 1828.52$) and mental state-based ($M = 4326.67, SD = 1376.35$) conditions; $t(20) = 2.635, p = 0.016$. These results show that the version the participant played has an effect on their score, more specifically that users reach higher scores playing the performance-based prototype compared to the mental state-based one. Therefore, we refute our hypothesis.

**V. DISCUSSION**

There are some important factors that may explain our results. One reason why this happened may be based on the limit values we used in the performance-based state machine.
Each level has its own definition for Poor, Medium and Excellent performance in it. In level 1, players had an Excellent performance if they killed at least 75% of the enemies that spawned in the last twenty seconds. We choose 75% to make it hard to jump from levels 1 to 2, since in level 1 players only had access to the regular weapons and no other interactions to kill enemies. We used a similar approach in level 2, where participants only have an Excellent performance if they kill 90% of the enemies. The value is higher because it is easier to kill them, since enemies now explode when they die, so naturally they have to kill a greater number to go from levels 2 to 3 than from 1 to 2. Also, we have to consider that in level 2 enemies spawn faster than in level 1. With this technique, we aim to prevent players from climbing the levels too fast. If they went up levels too fast, they might end with mismatched skills for the challenge and die in a short time. We considered that it would ruin the gaming experience and players wouldn’t be motivated to play well if they would end up dying quickly each twenty seconds. On the contrary, we decided to make it easier for the player to go down on levels. The principle is that, after players experiment the advantages of levels 2 and 3, they will focus on improving their performance to reach the next level. This way, if players are in level 2 and they kill less than 50% of the enemies or if they are in level 3 and kill less than 75% of the enemies, they go to level 1 and 2, respectively. Our first impression is that the values we used to define the Poor, Medium and Excellent were so well adjusted to lead the player to a higher level with more engagement elements that they ended up preferring the performance-based state machine instead of the one that tried to provide the best gaming experience by keeping a balance in challenge and skills.

The fact that our hypotheses were not accepted may be connected to some methodological limitations on our research that must be considered. When we collected data to model the classifier, the number of participants was small. A higher number of participants would allow to generalize for the whole population and take conclusions with a stronger impact. In future studies, a bigger sample size must be acquired. Also, the number of female participants was too small compared to the number of male participants, so future samples should consider increase female presence as well. These problems also showed up in the second testing phase, but with a smaller sample of participants and another disparate ratio of male and female gamers.

Moreover, our participants from the first testing phase did not cover the whole range of player expertise, as most players did not play the game at least once a week. Only in the first testing phase we found that the game periodicity influenced the anxiety in version C, yet, as it was only one case in sixteen possible cases, different gaming periodicity regarding FPS was assumed to be minimal. Concerning the second testing phase, the fact that the user played our game before or the game periodicity did not provide significant changes in the results.

Another limitation was the OpenSignals (r)evolution software. Since it did not allow a real-time recording, the assistant had to keep on restarting the recording after each twenty seconds. This was not practical and led to considerable small delays on the update of the file being processed by the classification framework. Nonetheless, these delays were rare or no longer than forty seconds, thus we consider them irrelevant.

Finally, we had to repeat the tests from the second testing phase because one of the sensors was not working properly and we only noted it after doing some signal processing.

Our findings provide additional evidence for inducing and investigating different mental states achieved while playing computer games. In particular, game developers may use the game design options to induce anxiety, boredom, flow and frustration or to maintain an optimal challenge based on the players mental state or performance. These mental states can be useful to provide a better gaming experience with task difficulty adjustment.

VI. Conclusions

This dissertation addresses the problem of these players becoming uninterested for not having their skills paired with the challenge of the game. In particular, we investigated if the mental state flow may be relevant for gameplay adaptability and may offer a better gaming experience, since that mental state is associated with deep engagement. In order to do so, we created an hypothesis supporting that a game that adapts to the gamer’s mental state following the flow theory provides a better gaming experience compared to a game that adapts to their performance. Although there were empirical evidences that users would have a higher flow state and scores playing a game that adapted itself to their mental state, after our testing phase, we found out that they have a higher flow state and scores playing a game that adapts to their performance. A deeper statistical analysis confirmed that players have a higher flow state and scores with a performance-based adaptable game, so in fact players do not have a higher flow state and scores with a mental state-based adaptable game. Hypotheses H1 and H2 addressed these topics and we refuted both, showing that participants had higher self-perceived flow and in-game performance in the performance-based adaptable game compared to the mental state-based adaptable game.

Our developed work resulted in a set of three main contributions: the game with adaptable parameters and environment, a classification framework that effectively detects the mental state of the player through biofeedback measures and a controller framework that effectively adapts a game based on the mental state of the player. We successfully created a classification framework that detects the player’s mental state in real time through biofeedback measures. Laine et al. [26] had an accuracy of 87% and Wilson and Russel [27] had mean classification accuracies of 85%, 82% and 86%. Our values are similar to their work, since the MLP had an accuracy as high as 89% with the validation set and 87% with the test set. We also contribute with an adaptable game design and the modeling of the controller framework. Both the game and the controller framework proved to effectively adapt the game engine depending on the on the feedback
from the classification framework. We consider these to be relevant contributions to the field of study, which will lead to the writing and submission of another scientific paper.

In the end, we were not able to accomplish our main goal which was to prove that a mental state-based adaptable game provided a better gaming experience to the player. In spite of it, we need further research to study our hypothesis in full, since we had a small number of participants and without our desired FPS gaming periodicity.

Future work involves creating a larger data set with a higher number of participants of both genders. This way we can generalize for the whole population and model a classifier with higher accuracy. We can also validate the plots created in Section III-A2, so that we can extend them to all the FPS genre. The number of participants that play FPS must be increased, since they are used to the game type and can provide a better feedback than players who only play mobile games, for example. Another approach is to complement our dataset with more representations of mental states of the user in order to provide a wider set of adaptability components and address more dimensions other than skills of the player and challenge of the game. Further development of the MLP classifier involves using different features from the ones we chose and choosing different physiological signals. Regarding software we used in our study, there is future work using the OpenSignals (r)evolution application programming interface to create a real-time recording of the physiological signals, which may prove to be a more effective way to access the participants’ physiological signals without the disruption of restarting the recording at each time interval.

REFERENCES