

Article

Towards Adaptive Gamification: A Method Using Dynamic Player Profile and a Case Study

Inmaculada Rodríguez ^{1,2,*}, Anna Puig ^{2,3,*} and Àlex Rodríguez ^{2,†}¹ Departament de Matemàtiques i Informàtica, UBICS Research Institute, UB, 08028 Barcelona, Spain² Departament de Matemàtiques i Informàtica, Universitat de Barcelona, Avda Corts Catalanes, 585, 08007 Barcelona, Spain; alexrodriguez@ub.edu³ Departament de Matemàtiques i Informàtica, IMUB Research Institute, UB, 08028 Barcelona, Spain

* Correspondence: inmarodriguez@ub.edu (I.R.); annapuig@ub.edu (À.P.)

† These authors contributed equally to this work.

Abstract: The design of gamified experiences following the one-fits-all approach uses the same game elements for all users participating in the experience. The alternative is adaptive gamification, which considers that users have different playing motivations. Some adaptive approaches use a (static) player profile gathered at the beginning of the experience; thus, the user experience fits this player profile uncovered through the use of a player type questionnaire. This paper presents a dynamic adaptive method which takes players' profiles as initial information and also considers how these profiles change over time based on users' interactions and opinions. Then, the users are provided with a personalized experience through the use of game elements that correspond to their dynamic playing profile. We describe a case study in the educational context, a course integrated on Nanomoocs, a massive open online course (MOOC) platform. We also present a preliminary evaluation of the approach by means of a simulator with bots that yields promising results when compared to baseline methods. The bots simulate different types of users, not so much to evaluate the effects of gamification (i.e., the completion rate), but to validate the convergence and validity of our method. The results show that our method achieves a low error considering both situations: when the user accurately ($Err = 0.0070$) and inaccurately ($Err = 0.0243$) answers the player type questionnaire.

Keywords: gamification; adaptive gamification; player types

Citation: Rodríguez, I.; Puig, A.; Rodríguez, À. Towards Adaptive Gamification: A Method Using Dynamic Player Profile and a Case Study. *Appl. Sci.* **2022**, *12*, 486. <https://doi.org/10.3390/app12010486>

Academic Editor: Aida Valls

Received: 15 November 2021

Accepted: 1 January 2022

Published: 4 January 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The use of game mechanics in non-gaming contexts, which is known as gamification, has come into play to encourage and motivate user's behaviors. The design of gamification should pursue a clear objective. For example, a gamified fair [1] was designed to motivate users to rate the experience of visiting a fair. Another gamified application was created with the goal of connecting students online and was designed to engage them in sharing documents and insights about their classes [2]. The gamification of a massive open online course (MOOC) aimed to foster students' engagement and therefore increase the completion rate of courses [3].

The design of gamified experiences usually adopts the one-fits-all approach, which may fail as a result of considering that all users have the same profile. An alternative strategy is adaptive gamification, which considers that users have different motivations during their interaction in gamified systems. Adaptive gamification is based on player type classifications. A pioneering, representative work on adaptive gamification was proposed by Lavoue et al. [4]. They gathered information about the user profiles (i.e., player types) at the beginning of the experience and then suggested to the users game elements that fit those (static) profiles. However, this initial player profile may either be inaccurately input by the user or may evolve slightly over time [5] and therefore, sometimes the suggested game elements do not completely correspond to the real profile. Additionally, adaptive gamified

systems usually take the most predominant player type to show the user game elements related to this predominant player type, with the drawback of ignoring other player types that may also characterize the user, so several player types should be considered, not only the predominant one [6].

In this paper, we extended our proposal of (dynamic) adaptive gamification presented in [7]. It used initial players' profiles, gathered from the HEXAD questionnaire [8], as well as information about users' interactions and while using the system. The goal of gamification was to foster users' engagement and thereby motivate the completion of online activities such as learning activities in a course or employees' progress report in a company. Concretely, we presented a case study focused on an STEM course integrated on Nanomoocs, a MOOCS platform. The case study first shows a real example of adaptive gamification in the educational field, and secondly, helps illustrate the elements that our method is based upon. Moreover, we depict the software architecture of the adaptive gamification system. Although evaluations of the effects of gamification are commonly carried out with users, this paper also presents an analysis of the proposed strategy using bots. The bots simulate different types of users, not so much to evaluate the effects of gamification (i.e., completion rate), but to validate the convergence and validity of our method. The results show that our method achieves a low error considering both situations: when the user accurately ($Err = 0.0070$) and inaccurately ($Err = 0.0243$) answers the player type questionnaire.

2. Previous Work

In this section, we first present research works concerning the data needed for the adaptation (player modeling), and then we focus on studies that describe the adaptation strategy: the context (educational systems, collaborative systems) and how the adaptation is carried out (static, dynamic).

2.1. Player Modeling

User-centered techniques have been proposed to correlate game elements and different user profiles [8,9]. Some of these techniques have focused on specific user characteristics such as motivation [10], personality traits [11,12], learning styles [13], player types [4], and types of interaction with different activities [14]. Others combined different characteristics, such as [15] which took into account learning styles and player types to determine the types of educational activities and game elements to include in a learning pathway. In contrast, ref. [16] suggested using the context, interactions, gender and player type to decide, through rules, the next game element to display. Other works instead focused on emotions to predict the individual's performance in gamified tasks, which is information that could potentially be used to adapt the game features [17].

Specifically, in current adaptive gamification approaches, the most commonly used taxonomies of player types are Bartle [18], BrainHex [19] and HEXAD [20]. These taxonomies allow us to easily identify player types from questionnaires as well as establish the correspondence between these player types and the game elements [9]. A recent study [21] proposed the so-called MoMo (motivational value model), a new prediction model that combines the four categorization models—Hexad, Bartle, Big Five and BrainHex. MoMo was validated along with Bartle and BrainHex using health applications. Results showed that it could predict players' preferences better than the individual models.

Questionnaires allow for the characterization of the player type of a user with a set of ratings. They determine the player type prior to the experience. For example, the HEXAD model distinguishes between six player types (achiever, player, philanthropist, disruptor, socializer and free spirit), and as result of the questionnaire, the user can obtain the following ratings: achiever 25%, player 18%, philanthropist 21%, disruptor 8%, socializer 10% and free spirit 5%. The decision of the final player type—and thus the most appropriate game element—usually relies on the predominant rating [4] or a combination of them [22]. As an alternative to questionnaires, a recent study [23] proposed to predict HEXAD player

types through gameful applications. The authors created two applications. An interactive one resembling a questionnaire but with appealing look and gameful feedback, and another application in which users interacted with gamification elements by shooting snowballs. The former correctly predicted player types but the latter did not show sufficient variance to reliably predict HEXAD player types. However, users interacted with gamification elements that fitted their player type, which seemed promising and worthy of further study.

In this work, we relied on the HEXAD player model and used the questionnaire as proposed and validated by [24]. However, it should be noted that in all the studies analyzed, the no-player type of player was not taken into consideration, a fact that suggests that, in some cases, the use of gamification is more detrimental than beneficial [25]. Thus, we also added the no-player player type to the HEXAD taxonomy, and we also considered for each user the assessment of all their ratings included in their player type.

Certainly, as supported by different studies [26], the interpretation of the results of these questionnaires may not be very reliable. Even if the questionnaire is valid, the answers may be somewhat random or the results at the beginning may not persist during the experience depending on the moment or the mood of the user. In fact, in addition to questionnaires, some proposals in the literature also gathered user feedback on learning activities [14], or scores on different game elements during the experience. Thus, inspired by these works, we enriched the user model obtained from the initial questionnaire by means of user interactions and opinions during the course of the activity. Therefore, we will base the adaptation of the game mechanics to the “real” and “dynamic” user profile using two types of interactions: on the one hand, the interactions with the game elements; and on the other hand, the opinions that the user can give at certain moments regarding those elements. Thus, using both types of interactions, we will refine the player types during the experience, and consequently, the game elements will be activated.

2.2. Adaptation Strategy

Recent literature reviews have analyzed adaptive gamification in educational and collaborative systems [8,27]. In the educational context, different studies linked game elements to students’ motivation and their player and learner type [11,13,28,29]. These studies laid the groundwork on which adaptation studies were constructed upon, which brought a variety of contributions from adaptation engine architectures [16,30] to evaluating gamification effectiveness [4,31,32]. The main needs of adaptive gamification systems emerging from the literature analysis in education are to enhance learners’ models, to explore different adaptation methods, especially dynamic adaptation, and to assess the long-term impact of gamification in learner performance or motivation.

Furthermore, gameful applications for motivating ones; participation in collaborative systems spread between serious games [33,34] and gamified experiences [35]. There were different approaches to adaptation in gamified collaborative systems. These include difficulty adaptation, which that either be based on the player behavior (player’s performance) [36,37] or based on the global behavior of a group of players [38], adaptive curriculum guidance [39,40], storytelling and content adaptation [41,42], adaptive presentation [43], and motivational interventions [32,44]. Our research work relates to the last two approaches as it presents to the user those game elements that fit their profile to motivate them to complete the course. Moreover, Ayastui et al. [27] formalized adaptation strategies and proposed a new taxonomy—gamification elements adaptation strategy (GEAS)—for the adaptation of gamification elements. The so-called full GEAS strategy refers to the adaptation that applies different gamification elements at different moments depending on the estimated user preferences [40,45]. Single GEAS adjusts some features of the gamification elements according to players’ behavior [32,46]. Our research situates in full the GEAS element of the taxonomy as our adaptation provides different game elements depending on the MOOC’s learners’ profile and their behavior along the course.

Nevertheless, there is still room for improvement regarding dynamic adaptation, where we find few studies. Lavoue [4] proposed a matrix factorization model similar to

those used in recommender systems. They used two matrices: one defining the player types of all users and the other representing how the game elements match the player types. They combined these two matrices to obtain game elements' scores for each user, and then they selected the element with the highest score. Another study [15] defined an off-line Q-Learning algorithm to generate an adaptive learning path for the user. The authors specified an S-Table and a Q-Table that represented the corresponding state at each taken action, and the Q-values of each action in each state, respectively. Both tables were the same for all the profiles. Nevertheless, the R-Table (the reward of each action in each state) was specific for each learning-player profile. All of them considered adapting game elements to the initial player's profile, keeping the player type static throughout the experience.

Our adaptive algorithm is based on a matrix factorization model similar to [4], which allows the recalculation of the player type, i.e., of all player ratings, during the experience in order to adapt the game element to the user model at any given moment. In this initial study, adaptation is based on activating one of the most appropriate game elements at any time depending on the recalculated player type.

3. Runtime Method for Adaptive Gamification

In this section, we present our proposal for adaptive gamification. First, we introduce the main definitions our method is based on: (i) the HEXAD player type model, including the non-player type; (ii) the selected game elements as a subset of those proposed by Tondello [24]; and (iii) we defined the matrices and vectors of ratings and interactions implied in our algorithm. Second, we describe how we match these player types with their corresponding game elements using an extended matrix factorization method [4].

3.1. Previous Definitions

3.1.1. Player Type

An essential concept for adaptive gamification is the player type model, which classifies what kind of game elements maximizes user motivation. As we mentioned above, we based ourselves on the HEXAD player typology, adding the non-player type [20]. Player types are defined as

$$PT = \{pt_1, pt_2, \dots, pt_7\} \quad (1)$$

The player types pt_i are explained below:

1. Disruptor: motivated by the ability to modify the system;
2. Free spirit: motivated by the ability to freely explore the system;
3. Achiever: motivated by the ability to win challenges and unlock hidden content.;
4. Player: motivated by the game itself;
5. Philanthropist: motivated by the ability to share goods and help other users;
6. Socializer: motivated by social connections;
7. Non-player: users who do not like to play.

The player type of a particular user is represented as a vector of length 7, PR , where each component r_i represents its ratings for each player type, so the values vary between 0 and 1, $PR = (r_1, r_2, \dots, r_7)$. For example, a user can be 20% disruptor, 10% free spirit, 30% achiever, 40% player, 40% socializer, 10% philanthropist and 0% no-player, which is encoded with the vector (0.2, 0.1, 0.3, 0.4, 0.4, 0.1, 0). Since the user's player type changes over time, we define $PR^{(t)}$ as the player ratings at the time t as

$$PR^{(t)} = (r_1^{(t)}, r_2^{(t)}, \dots, r_7^{(t)}) \quad (2)$$

3.1.2. Game Element

Based on the correlation analysis of the HEXAD player types with 52 game design elements performed by Tondello et al. [24], we selected a subset of 14 game elements (see Table 1) covering the whole spectrum of player types (see Figure 1). The 52 game design elements were grouped by player types using the correlation value of each player type's

mean score and the corresponding design elements' mean score per user. In our study, considering the fact that all game elements are equally motivating for one player type, we selected at least two of those: one related to its nature itself (for example, the Easter Egg game element is specific to the free spirit players and not others), and another considered complementary or additional for another type of player, if it exists (for example, the social status is specific to the socializer but is also an additional game element for player types such as disruptor or player). It should be noted that in the analysis carried out by [24], they did not find game elements correlated to the philanthropist player type. Then, we opted to include two of the philanthropist game mechanics suggested by [20] (knowledge sharing and gifting). There are no game elements associated to non-player type. Moreover, the educational context of our application further helped us either select or discard the game elements for each player type. For example, a development tools mechanism can be implemented as a prize that allows disruptors to change the system, e.g., change the design of badges. Nevertheless the anarchic gameplay mechanism of disruptors is not adequate for a massive online open course because its implementation can go against the learning objectives.

Game elements are defined as

$$GE = (ge_1, ge_2, \dots, ge_{14}) \tag{3}$$

The game elements, ge_i , are briefly explained below:

1. Development tool: allows the player user to create certain gamification mechanics such as badges, challenges and unlockables.
2. Challenge: the player must overcome a challenge, such as reaching a certain level and solving a problem in a certain time;
3. Easter egg: the mechanism consists of an image which, when pressed five consecutive times, allows access to a mini-game;
4. Unlockable: when a player overcomes a certain challenge, a hidden content is unlocked, which can be a message, a mini-game, etc.;
5. Badge: awarded to the player when they manage to complete a difficult task;
6. Level: shows the user's progress in completing a task, subdivided into levels;
7. Point: the player gains score, experience, virtual money, etc.;
8. Leaderboard: displays a ranking of scores;
9. Gift Opener: the player opens gifts they have received;
10. Lottery: game of chance (roulette) that allows players to increase their scores;
11. Social network: a small social network that allows players to create a profile, add friends and view their profiles;
12. Social status: collection of rankings of players based on their scores, especially those related to social interactions, such as the number of followers, visitors, etc.;
13. Share knowledge: the player sends help messages to everyone in a group;
14. Gift: the player sends gifts to other users.

It is worth mentioning that each game element does not target only one type of player but can motivate different types of players. Thus, the i -th game element, ge_i , has associated a vector of motivation indexes GM_i , where each component, m_j , is the percentage of motivation it can cause in one of the seven types of players, p_j . For example, the fifth game element "Badge" (see Figure 1) can motivate users with both achiever (pt_3) and player (pt_4) player types, then the $GM_5 = (0, 0, 0.5, 0.5, 0, 0, 0)$:

$$GM_i = (m_1, m_2, \dots, m_7) \quad \forall i = 1 \dots 14 \tag{4}$$

Therefore, we define the matrix, M as

$$M = (GM_i)_{1 \leq i \leq 14} = (m_{ij})_{1 \leq i \leq 14, 1 \leq j \leq 7} \tag{5}$$

where the rows of this matrix are indexed by the game elements, and the columns by the player types. Thus, $m_{i,j}$ represents the percentage of motivation (motivation index) that the i -th game element produces for the j -th type of player.

Finally, as can be appreciated on the right-hand side of Figure 1, the utility of using a game element, $U^{(t)}$, can be computed from the matrix M and the player type ratings, $PR^{(t)}$.

Table 1. Table of the distribution of game elements, ge_i , according to their primary player types and their additional player types.

Player Type	Game Elements	Additional Player Types [24]
Disruptor	Development Tools Creativity Tools (Challenges)	Free Spirit Player, Achiever, Free Spirit
Free Spirit	Unlockable Easter Egg	- Player
Achiever	Badge Level of Progression	Player Player
Player	Lottery Leaderboard Gift Opener (Prizes) Points	- - - -
Socializer	Social Network Social Status	Free Spirit -
Philanthropist	Share Knowledge Gifting	- -

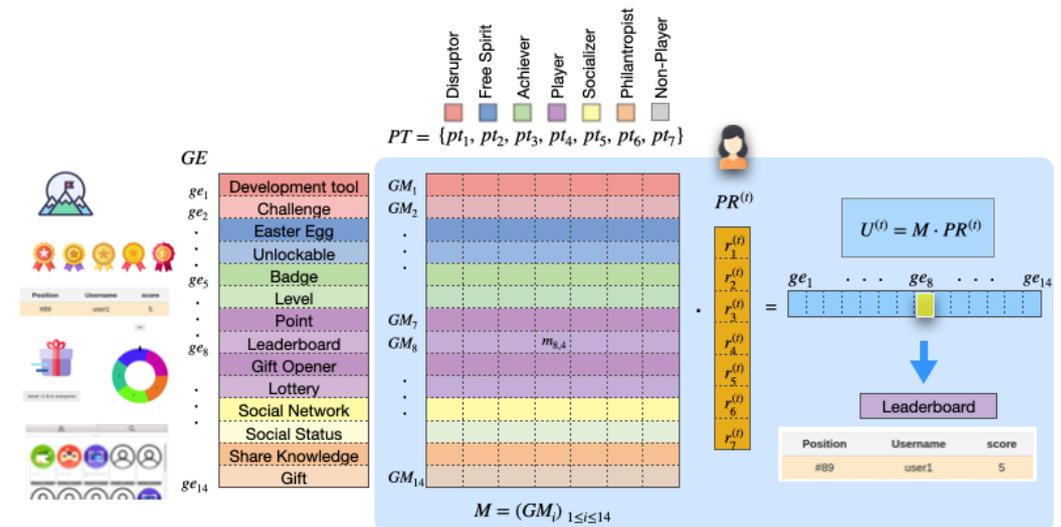


Figure 1. Fourteen selected game elements, ge_i , enumerated by $i = 1, \dots, 14$. Each color represents one player type, pt_j , ($j = 1, \dots, 7$). The vector $PR^{(t)}$ defines the player ratings of a user and the matrix M stores all the motivation values that the i -th game element produces for the j -th player type. The game element with the highest utility is the item that will be presented to the user. As an example, the game element 8-th, the leaderboard, is highlighted to represent the next game element to be shown to the user.

3.1.3. Interaction Index

Our method uses the so-called interaction index to determine whether a game element motivates the user. We define it as the percentage of the user’s interaction with each game element at time t . This is represented by a vector of length 14 (the number of game

elements), $S^{(t)} = (s_1^{(t)}, s_2^{(t)}, \dots, s_{14}^{(t)})$. The interaction index of the i -th game element, $s_i^{(t)}$, is defined by

$$s_i^{(t)} = 1 - e^{-\left(o_i^{(t)} \frac{n_i^{(t)} - n_i^{(t-1)}}{\tau_i^{(t)} - \tau_i^{(t-1)}}\right)} \tag{6}$$

where

$\tau_i^{(t)}$: the display time, i.e., the time interval for which the game element has been displayed until time t ;

$n_i^{(t)}$: the number of interactions at time t ;

$o_i^{(t)}$: the opinion, i.e., the user assessment of the game element. This is a value between 0 and 1. Opinions from 1 to 5 stars correspond to 0.2, 0.4, 0.6, 0.8 and 1, respectively.

Note that Equation (6) encodes the interaction index as a number between 0 and 1. If there are no interactions between time t and $t + 1$, the interaction index is 0 (since $(n_i^{(t)} - n_i^{(t-1)})$ is 0). The interaction speed is $\left(\frac{n_i^{(t)} - n_i^{(t-1)}}{\tau_i^{(t)} - \tau_i^{(t-1)}}\right)$. Then, the interaction index tends to 1 as the interaction speed increases.

The opinion $o_i^{(t)}$ modulates the interaction speed: $s_i^{(t)}$ tends to be faster than 1 when $o_i^{(t)}$ is near 1 than when $o_i^{(t)}$ is near 0.

3.1.4. Activities

Considering that any serious context to be gamified is composed of activities (e.g., exercises, videos, readings), game elements unfold as a result of users performing n activities:

$$A = \{a_1, a_2, \dots, a_n\} \tag{7}$$

Then, users' experience in the system consists of a series of activities with intertwined gamification elements: $a_1, a_2, ge_i, a_3, a_4, a_5, ge_j, \dots, a_n$, where $ge_i, ge_j, \dots \in GE$. The timing of ge_i appearing in this sequence of activities is actually defined by the gamification designer, who is the teacher in the case of MOOCs. Note that the teacher receives some recommendations regarding how to alternate activities and gamification elements in the course. For example, adding game elements after hard learning activities, such as reading a long text, is highly recommended, while in more entertaining activities, such as infographics, the subsequent use of game elements is not as essential.

3.2. Adaptive Method

Our method relies on the fact that the participant's inner player type (or real player type, RPT) may evolve slightly (i.e., is dynamic) during the experience. We therefore assume that this inner profile at the start of the experience, RPT_0 , can be approximated by an initial player type, $PR^{(0)}$, using known and validated questionnaires [20]. Furthermore, we assume that the users' behavior—interactions with and opinions of game elements—reveal their real player type, RPT^t . Thus, we will take into account this behavior to approximate a different $PR^{(t)}$ during the experience. To recompute the player type in each iteration t , our algorithm takes into account the current $PR^{(t)}$ as well as the interactions that users made with the different game elements and how they rated them.

Once the type of initial player is defined, our method iteratively updates the player profile of the user at time t , $PR^{(t)}$, and thus calculates the utility of showing one or another game element to the user, $ge_i^{(t)}$. Each iteration consists of the three steps depicted in Figure 2 (considering the definitions introduced in Section 3.1). The steps are: (1) obtain $PR^{(t+1)}$; (2) score the utility of showing a game element to a user at a specific time $t + 1$ (denoted by $U^{(t+1)}$); and (3) select which game element to activate based on the assigned scores.

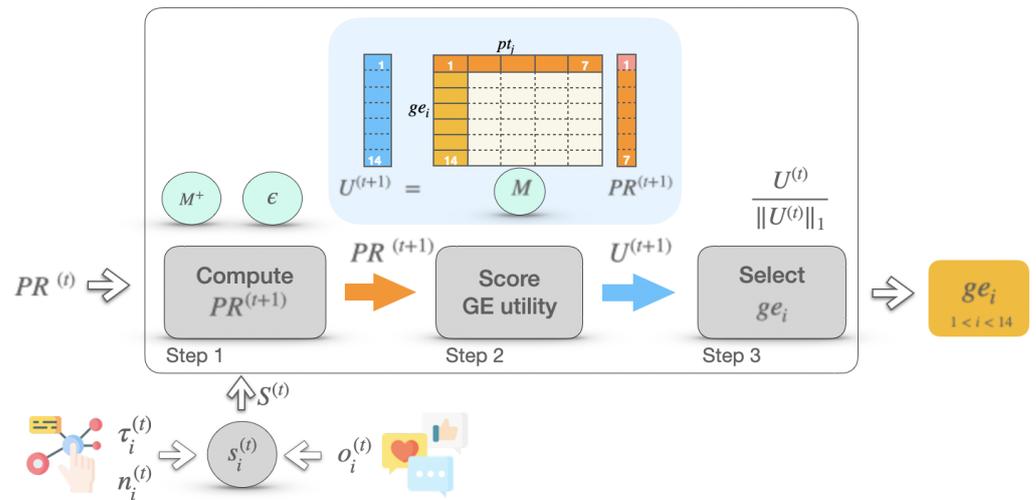


Figure 2. Steps to compute the utility of showing a ge_i associated to a user at time $t + 1$. Blue circles indicate the constant data of our method. All other elements define dynamic values that change over time.

In the first step, we compute the new player type ratings, $PR^{(t+1)}$:

$$PR^{(t+1)} = (1 - \epsilon) PR^{(t)} + \epsilon (M^+ \cdot S^{(t)}) \tag{8}$$

where

$PR^{(t)}$: the player type of the user at time t ;

$S^{(t)}$: the interaction indexes;

ϵ : to avoid extreme fluctuations between $PR^{(t)}$ and $PR^{(t+1)}$, where $0 < \epsilon < 1$ —the value of this parameter should be tuned experimentally;

M^+ : the Moore–Penrose pseudoinverse matrix of M , needed in order to interpret $S^{(t)}$ and $PR^{(t)}$ in the same space.

In the second step, once we calculated the new player profile, we compute the utilities as indicated in the top right-hand side of Figure 1, using the matrix M defined in Equation (5):

$$U^{(t+1)} = M \cdot PR^{(t+1)} \tag{9}$$

Finally, in the third step, we select the next game element to display considering the i th component of $\frac{U^{(t)}}{\|U^{(t)}\|_1}$ as the probability of choosing the i th game element using a weighted random choice ($\|\cdot\|_1$ is the ℓ_1 -norm).

4. Case Study

4.1. Adaptive Gamification in Nanomoocs

Nanomoocs is an innovative massive open online course (MOOC) platform consisting of online courses designed as training pills. They are focused on well-identified user segments, with a systematized instructional design, high-quality audiovisual content, and incorporating technologies to improve learning such as peer review, personalization, gamification and emotions recognition. Concretely, our case study of adaptive gamification focuses on the course entitled “What can we do with the plastics in the sea?”, intended for secondary school students aged between 14 and 15 years.

The course aims to develop the critical thinking and problem-solving skills of students regarding the topic of plastics in the sea. Students must follow a scientific method and reason about how to solve the problem of plastic discharges into the sea. They have to look for, contrast and select information sources and digital media in order to build new knowledge on the problem of plastics in the sea. They will analyze in a critical way

how plastics affect the life of living beings, find out about and analyze different forms of extraction and transformation of plastic, and finally carry out a decision-making process in order to propose preventive and solution measures. The course consists of four formative units, each containing activities (A_i) such as videos, interactive info-graphics, readings and questionnaires (see Figure 3).

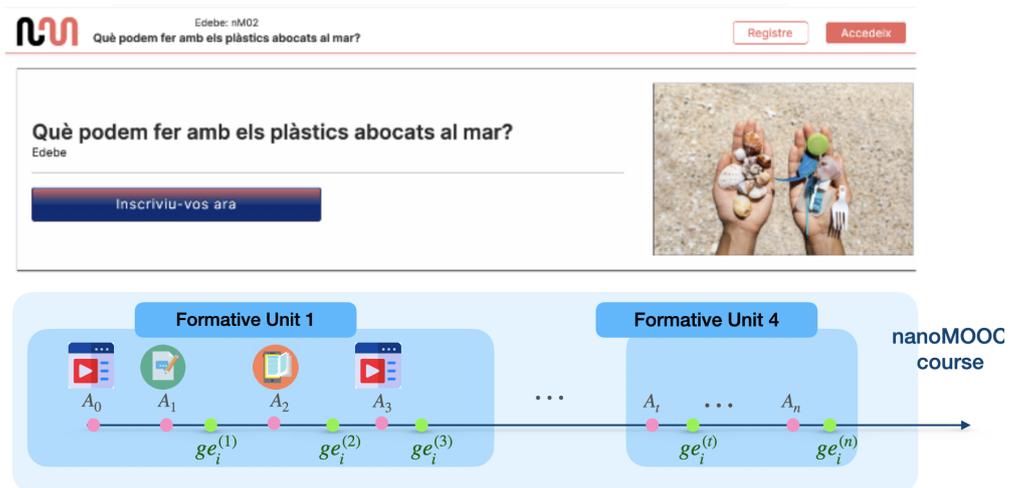


Figure 3. Nanomooocs about plastics in the sea.

Gamification elements (ge_i) appear when the participants complete a number of activities. For example, when students complete an interactive infographics and a posterior questionnaire and just after the completion of the questionnaire a gamification element (GE) unfolds to motivate and reward the student for the work performed. As the gamification system is adaptive, different students will be awarded with different gamification elements depending on their player type (see Figure 4).

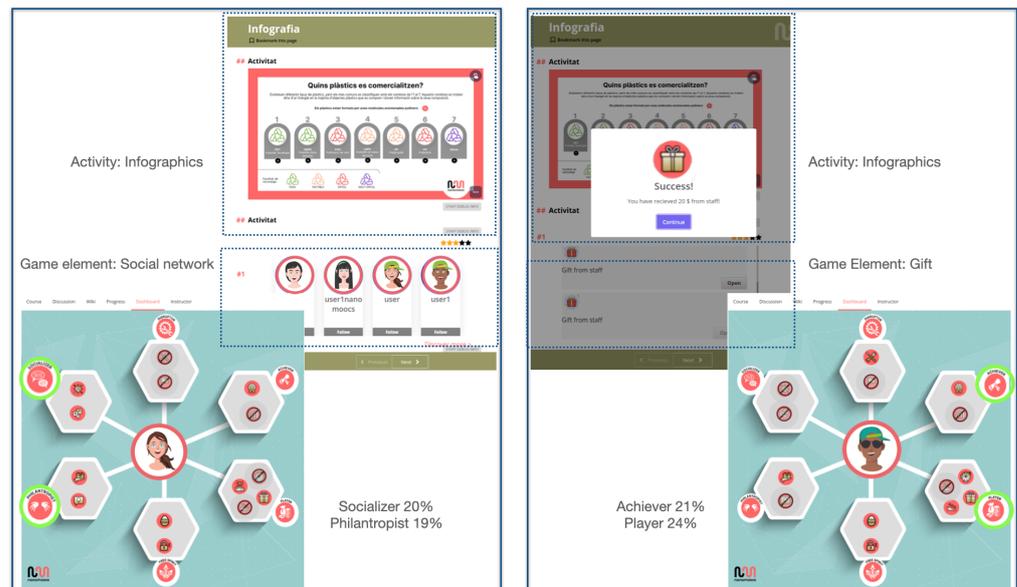
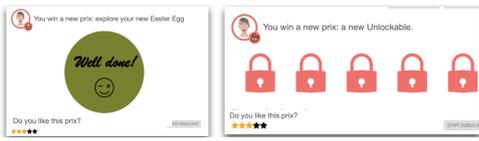
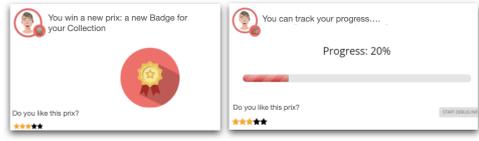
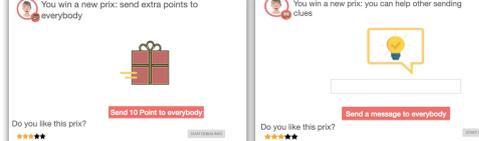
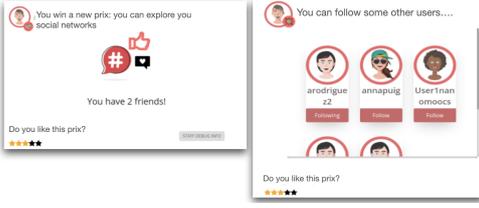
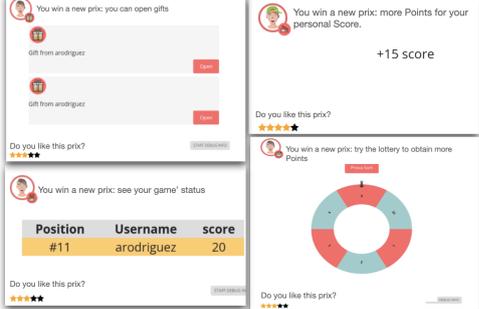


Figure 4. Two game scenarios exist for the same activity: on the left, the enabled game element for a socializer player; on the right, the activated game element for an achiever/player user.

The adaptive system needs an initial player profile of the participant which is gathered through a player type questionnaire (<https://www.gamified.uk/UserTypeTest2016>, accessed on 1 December 2020) [20] that the students answer at the beginning of the course. The result of the questionnaire is the initial information the system has about the participant,

i.e., the way they like to play. The information gathered by this questionnaire has been referred to before in this paper as $PR^{(0)}$ —player ratings at time 0. The gamified system then incorporates this profile in a dashboard that shows each user their player type (see Figure 4 and Table 2).

Table 2. The six game scenarios related to the activation of the fourteen game elements.

 <p style="text-align: center;">Disruptor</p>	 <p style="text-align: center;">Free Spirit</p>
 <p style="text-align: center;">Achiever</p>	 <p style="text-align: center;">Philanthropist</p>
 <p style="text-align: center;">Socializer</p>	 <p style="text-align: center;">Player</p>

Note that we face two different scenarios when asking participants to fill in the player type questionnaire. The first scenario is when the participant fills in the questionnaire conscientiously and thoroughly and the adaptive system then uses an initial player type data, $PR^{(0)}$, which is approximately equal to the real player type RPT_0 , i.e., $RPT_0 \simeq PR^{(0)}$, as can be seen in the top of Figure 5. The second scenario occurs when the participant answers the questionnaire in an unreliable and untrustworthy way, i.e., the results of the questionnaire, $PR^{(0)}$, are far from the RPT_0 , i.e., $RPT_0 \not\approx PR^{(0)}$, as can be seen in the bottom of Figure 5.

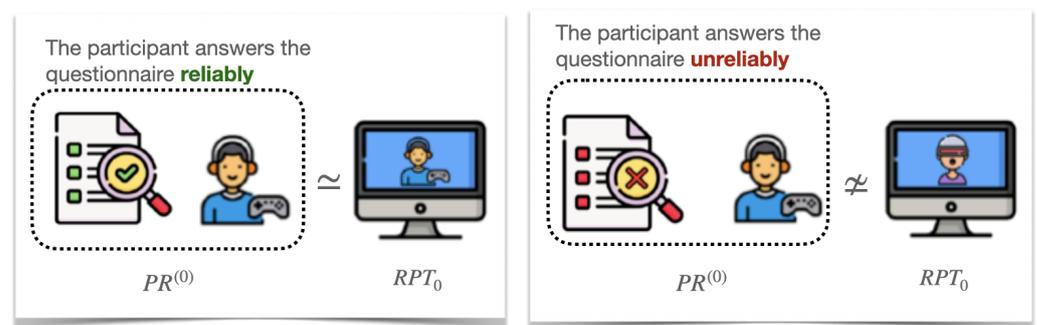


Figure 5. Player type questionnaire results as $PR^{(0)}$ (player type ratings at time 0) versus real player type RPT_0 .

We also consider the case of participants that do not want to play (non-player type), and consequently do not answer the questionnaire. In this case, we have no initial information about the player type profile, but the system gives the user the opportunity to join in the gamified experience, when the user shows any interest in the random game mechanics that will appear during the experience. If there is no interest, the system will cancel the adaptive gamification for the non-player user.

4.2. Adaptive Gamification: Software Architecture

Our system is based on a service-oriented architecture in which the gamification system resides on an external server as it is shown in Figure 6. In our case, the learning management system edX (<https://www.edx.org>, accessed on 1 November 2021) hosts the Nanomoocs course and uses the gamification service through a restful API.

The edX platform uses so-called XBlocks as basic units to define activities within the course units. Thus, a specific gamification XBlock was created to encapsulate the calls to the external gamification API to ask for the next game element $ge_i^{(t)}$ to be displayed once the user completes the activity. Moreover, that XBlock is in charge of displaying the game element as well as monitoring the user's behavior with this game element. It captures the user's interactions $n_i^{(t)}$, the display time $\tau_i^{(t)}$ and the user's opinions $o_i^{(t)}$ that the user realizes in the game element. Every 15 s, all these interactions are sent to our adaptive method, running in an external gamification module (see the right part of Figure 6), to update the player profile, $PR^{(t)}$. This means that the iteration frequency of our method is approximately 15 s in an attempt to have enough time to gather information about changes in user behavior without overloading the system.

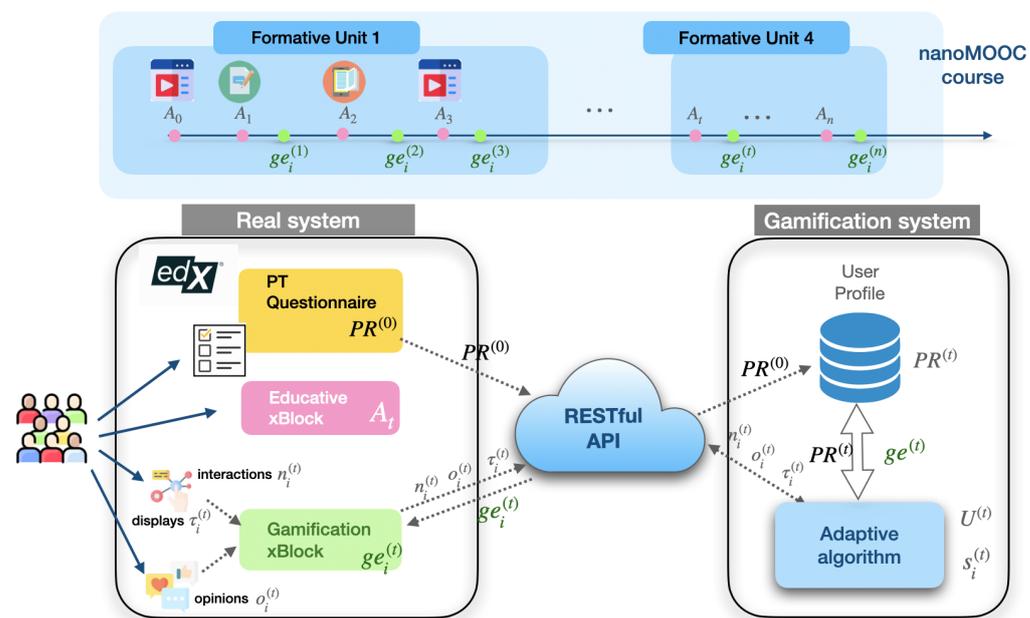


Figure 6. Software architecture to extend edX with adaptive gamification.

Therefore, when users access the course, they fill in the player type questionnaire (integrated within the course as another possible activity) and this first player rating, $PR^{(0)}$, is sent to the gamification server. From this point on, the user performs learning activities within the learning management system (LMS) and when some of them are finished, an event calls to the gamification service using the gamification XBlocks (see the =Nanomoocs timeline in Figure 3). At this point, edX obtains the next game element to be displayed, $ge_i^{(t)}$, according to the current player ratings, $PR^{(t)}$. This game element will be displayed in the edX course via an HTML file, which also contains JavaScript calls to the API to monitor the user's interactions, and the display time they spend on that game element. The users

are also able to give their opinion about the active game element. Finally, the displayed game element will be enabled in the user’s dashboard. Later on, the user will be able to consult all the game elements that they have already activated. Furthermore, the dashboard is an HTML file that performs calls to the gamification service to update user information at any time.

5. Adaptive Method Evaluation

5.1. Simulation System

We assume that an adaptive gamification strategy works if the game element proposed to the users fit their “real” player profile at any time t . Considering that we simulate the player using a bot, in the following, we note the “real” profile of the bot as RPT_0 , and its player type rating at time t as $PR^{(t)}$. We also assume that the bot’s real player type does not change over time, $RPT_t = RPT_0$, in such a way that we can measure the convergence of our algorithm to a particular player type.

The values of RPT_0 come from a dataset containing user types HEXAD test results (<https://gamified.uk/UserTypeTest2016/user-type-test-results.php>, accessed on 1 November 2021), where 42,782 tests were carried out, obtaining average type scores for all the modalities of players. We selected the eleven most representative modalities (see Table *Summary* in previous URL) and their corresponding average type scores (see also Table *Average Type Scores*). For instance, the Achiever modality appears in 12% of tests and its average scores are $RPT_0 = (0.12, 0.18, 0.20, 0.16, 0.16, 0.17, 0.0)$ meaning 12% disruptor, 18% free spirit, 20% achiever, 16% player, 16% socializer, 17% philanthropist, and 0% non-player.

Bearing in mind that a real user would either reliably (accurately) or unreliably (inaccurately) answer the HEXAD questionnaire at the beginning of the gamified experience, our bot simulates users’ responses accurately or somewhat randomly. Therefore, we define $PR^{(0)}$ as being close to RPT_0 ($RPT_0 \simeq PR^{(0)}$) or far away from ($RPT_0 \not\simeq PR^{(0)}$). To do so, if the reliability is low, the bot rating $PR^{(0)}$ is the furthest non-null scores from RPT_0 in Table *Average Type Scores*. Otherwise, if the reliability is high, the bot takes its $PR^{(0)}$ as RPT_0 . Note that when the bot simulates accurate responses to the questionnaire, it is desirable that the value of $PR^{(t)}$ remains close to RPT_0 , while when it simulates inaccurate answers, it is convenient that $PR^{(t)}$ converges to RPT_0 when $t \rightarrow \infty$. Figure 7 shows the real profile of the bot and its initial player type rating, $PR^{(0)}$, computed far from or the same as RPT_0 in function of whether the questionnaire was reliably (accurately) or unreliably (inaccurately) “answered” respectively. Note that this figure is the counterpart of Figure 5 for the simulations using bots.

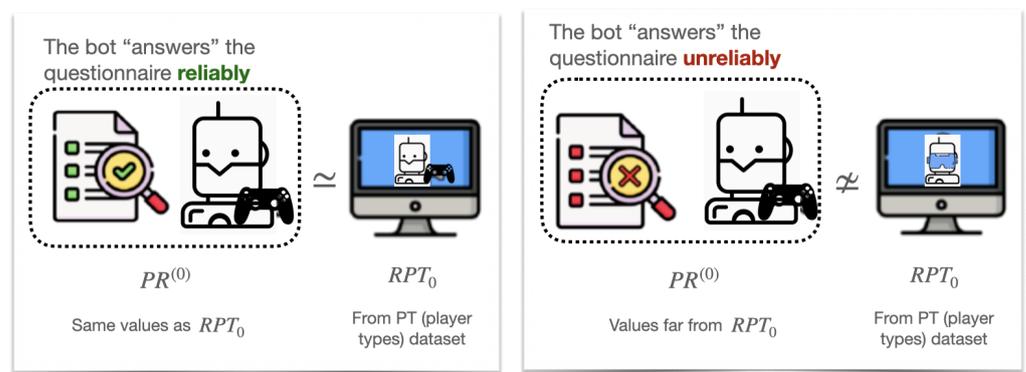


Figure 7. Bot’s player type initial rating, $PR^{(0)}$, versus real bot’s player type.

The bot simulates the interaction with the game elements (recall interaction indexes, $S^{(t)}$, and $\tau_i^{(t)}, n_i^{(t)}, o_i^{(t)}$ as input data of $s_i^{(t)}$ in the bottom left of Figure 2, and Equation 6).

To do so, we used two variables: the time between two consecutive interactions, $\top_i^{(t)}$, and the opinion, $\Theta_i^{(t)}$, defined as

$$\top_i^{(t)} = -9.5(GM_i \cdot RPT_0) + 10 \tag{10}$$

$$\Theta_i^{(t)} = \frac{1}{5} \text{round}(4(GM_i \cdot RPT_0) + 1) \tag{11}$$

where i corresponds to the game element (ge_i) selected by the method. Thus:

- If the game element (ge_i) fits the real bot's profile (RPT_0), the bot interacts more frequently than otherwise. Therefore, $\top_i^{(t)}$ reflects this behavior by taking values from 0.5 (frequent interactions) to 10 (longer time between interactions);
- Regarding $n_i^{(t)}$, we consider that the bot interacts once every $\top_i^{(t)}$;
- The opinion can be calculated in a similar way using $\Theta_i^{(t)}$.

Figure 8 shows the main characteristics of the simulation system. Note that it is the counterpart of Figure 6, where edX and the real user are substituted by the bot generation system and their simulated behaviors, while the gamification system remains unchanged. That is, the bot's behavior and the adaptive algorithm work independently; therefore, the bot behaves exclusively in function of its player profile. The main differences in Figure 8 are: (1) the initialization of the $PR^{(0)}$ is performed by the most common cases of player types, which are stored in a database; (2) the real bot player type is defined from these initial player types (RPT_0); and (3) the inputs of our method, i.e., $PR^{(t)}$ and users' interactions—their game element display times and opinions—are based on the bot real player type (RPT).

Regarding the parameters of our method, we settled the ϵ value to 0.001. As mentioned above, we define this ϵ to compute the new player type ratings at each iteration. Setting ϵ to a low value, we achieve low fluctuations between two consecutive player types without losing the influence of the interactions and the opinions of the user (see Equation (8)). Moreover, to perform the simulations as close as possible to the real case with users, we generate interactions and the bot's opinions every 15 s to update the player type, although the execution time of each iteration is lower.

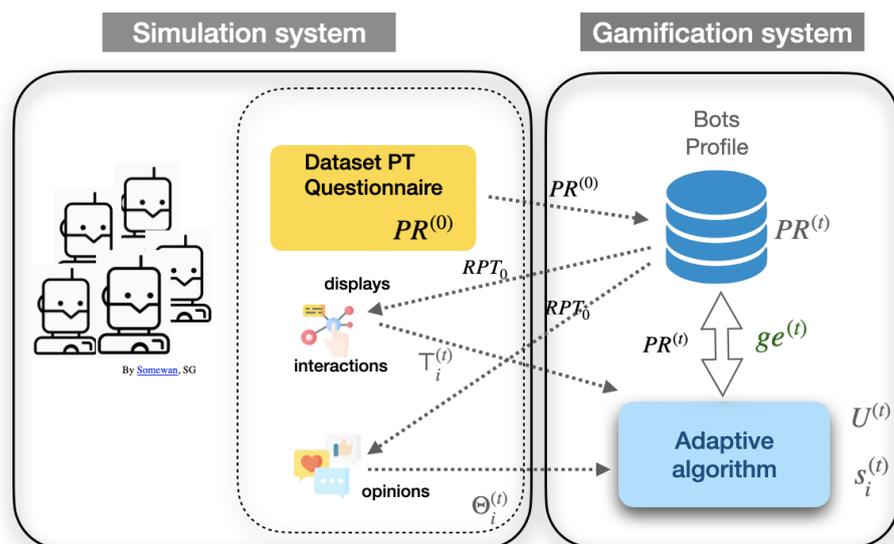


Figure 8. Evaluation infrastructure: on the left, the simulation system of bots; and on the right, the gamification system that takes bots' profiles instead of users' profiles.

5.2. Results

Once how the bot simulates a real user has been stated, we will consider three methods in function of whether the value of $PR^{(t)}$ is fixed or variable along the time (*A—constant player rating; B—random dynamic player rating; and C—our method: dynamic player rating*), assuming that the bot has previously answered the HEXAD questionnaire (fixing $PR^{(0)}$). Moreover, we define how the error is computed to compare the three methods. For each method, we analyze two cases (accurate answers, inaccurate answers).

1. **Method A**—Constant $PR^{(0)}$: a constant player rating, $PR^{(t+1)} = PR^{(0)}$, is assigned at any time. In this case, we use Equation (9) to calculate $U^{(t+1)}$ once and always use its maximal component.

We calculate the error as follows. Since $PR^{(t+1)} = PR^{(0)}$, the distance is simply calculated as $Err = |RPT_0 - PR^{(0)}|$. Note that if the user has accurately answered the questionnaire, we have $Err = 0$;

2. **Method B**—Random dynamic $PR^{(t)}$: a dynamic player rating $PR^{(t+1)}$ is randomly chosen at each time t and then, the game element selected to be shown is also the maximal component of $U_i^{(t+1)}$. Note that this method is equivalent to picking a random game element.

In this method, we compute the error as the average distance between RPT_0 and a random point $p \in \{(x_1, \dots, x_7) \in [0, 1]^7 : \sum_{i=1}^7 x_i = 1\}$ using the average distance of random points in a unit hypercube (average distance of random points in a unit hypercube, <https://martin-thoma.com/curse-of-dimensionality/>, accessed on 1 November 2021).

3. **Method C**—Our method, dynamic $PR^{(t)}$: a dynamic player rating $PR^{(t+1)}$ is computed according the $s_i^{(t)}$ defined by Equation (6). The bot then simulates $\tau_i^{(t)}$, $n_i^{(t)}$, and $o_i^{(t)}$ using $\Upsilon_i^{(t)}$ and $\Theta_i^{(t)}$. The game element selected to be shown is a weighted random choice of $U^{(t+1)}$ (see step 3 in Figure 2).

In this method, we calculate the error Err based on the distances between $PR^{(t)}$ and RPT_0 for all t from 1 to n_iter :

$$Err = \frac{1}{n_iter} \sum_{t=1}^{n_iter} |RPT_0 - PR^{(t)}| \tag{12}$$

Experiments were performed using the gamification software described in Figure 8 and run on a Windows 10, Intel Core i7 processor with 8GB RAM. Table 3 shows the mean errors (i.e., the distance between $PR^{(t)}$ and the real player profile RPT_0) and the standard deviation (SD) obtained by the three methods: (A) constant player ratings; (B) random dynamic player ratings; and (C) our method dynamic player ratings. In the simulations, the initial setup of a bot consists of its RPT_0 “real” player type, i.e., what define its playing behavior, and its approximated player type PR^0 , i.e., the results of its player type questionnaire. We analyzed the three methods on two cases (accurate and inaccurate answers). Moreover, we define the worst and the best scenarios in all methods.

The *worst scenario* is defined by the bot that produces the biggest error in method C. Similarly, the *best scenario* is defined by the bot that produces the smallest error in method C. Then, Table 3 also depicts the errors obtained by those bots in methods A, B and C that started the simulation using the same RPT_0 and PR^0 as the bots in the best and the worst scenarios of C. In method A, as we take into account $PR^{(t)} = PR^{(0)} = RPT_0$, accurate answers are the ideal case ($Err = 0$). However, with inaccurate ones the mean error grows to 0.0311. Moreover, both cases of method B have similar errors ($Err = 0.08024$ and $Err = 0.08027$ in accurate and inaccurate answers, respectively), because there is a random selection of player type ratings and thus, they are almost independent of the reliability (accurate or inaccurate) of the answers. Finally, except in method A with accurate answers, method C behaves better than A and B ($Err = 0.0070$ and $Err = 0.0243$ in accurate and inaccurate answers, respectively) even in its worst scenario ($Err = 0.0333$). Note that the

mean errors should be interpreted in the interval [0, 0.2857] since the maximum distance between two normalized points in a Unit Hypercube is 2/7 (approximately 0.2857). This indicates, for example, that the value of the error 0.0827 in case B-Random represents approximately 28%.

Table 3. Table of mean errors of methods A-Constant Player Type, B-Random Dynamic Player Type, and C-Our Method-Dynamic Player Type, when the questionnaire is accurately and inaccurately answered (the two cases in rows), together with the best and worst errors obtained with method C .

Cases/Methods	A—Constant Constant $PR^{(0)}$	B—Random Dynamic $PR^{(t)}$	C—Our Method Dynamic $PR^{(t)}$
Accurate Answers Case: $RPT_0 \simeq PR^{(0)}$			
Mean Error (SD)	0	0.08024 (0.00052)	0.0070 (0.00166)
Worst Scenario of C	0	0.0804	0.0105
Best Scenario of C	0	0.0797	0.0029
Inaccurate Answers Case: $RPT_0 \not\simeq PR^{(0)}$			
Mean Error (SD)	0.0311(0.00404)	0.08027 (0.00040)	0.0243 (0.00475)
Worst Scenario of C	0.0367	0.08012	0.0333
Best Scenario of C	0.0233	0.08018	0.0146

Regarding the results of the mean error in inaccurate answers, the difference between method A (Constant Player Type) and method C (Dynamic Player Type) is 0.0068. To analyze this difference, we conducted a statistical study. As the data are not normally distributed (Shapiro–Wilk p -value: 0.000005421, with an effect size large $ES = 0.239$), we used the Wilcoxon signed-rank test for paired samples. Results indicate that we can reject the null hypothesis ($mean_A = mean_C, p < 0.0001, ES = 0.87$) and then the difference is significant.

On the other hand, the player profiles of the worst (0.0333) and best (0.0146) cases of C in inaccurate answers were disruptor in both cases. Indeed, the disruptors, free spirit and philanthropists/achievers player types represent the most frequent results from the Marczewski questionnaire.

Figure 9 shows the values of a bot’s player ratings $PR^{(t)}$ along 180 iterations for the best case of method C in inaccurate answers, as shown in Table 3. The real player type, RPT_0 , is represented by dotted lines and colored stars (red—disruptor; green—free spirit; yellow—philanthropist) at the end of the X axis. As can be appreciated, the disruptor player type is the one that achieves the lowest error (distance between the red star and the end of the red line). The first values, 0.14 and 0.21, of the two initial vectors ($PR^{(0)} :: [0.14, 0.2, 0.17, 0.16, 0.15, 0.17, 0]$ $RPT_0 :: [0.21, 0.17, 0.16, 0.16, 0.14, 0.15, 0]$), and the final value obtained, 0.21, in the vector $PR^{(180)} :: [0.21, 0.185, 0.156, 0.149, 0.136, 0.162, 0]$, are the vector of player ratings in the following order :: (disruptor, free spirit, achiever, player, socializer, philanthropist, non-player). Additionally, not only does the predominant player type converge to its real value, but other player types that are not predominant, such as the free spirit rating (initially at 0.20), also evolve by slightly decreasing their values with regard to the real one (0.17).

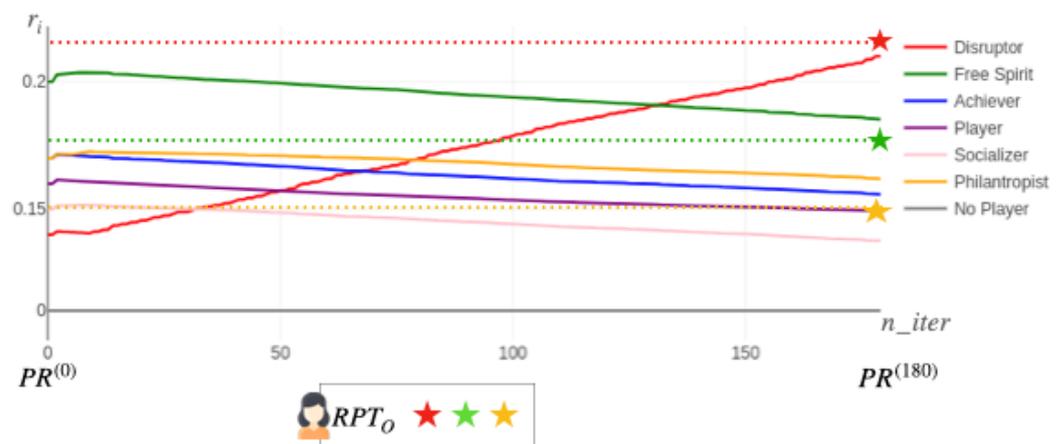


Figure 9. Evolution of the player rating in the best case in inaccurate answers (method C): philanthropist (yellow); free spirit (green); and disruptor (red).

6. Discussion

The results of our study conclude that the method *A*—where player type ratings are fixed along the experience—is the best when the users thoroughly and accurately answer the questionnaire, while our method, method *C*, works well in both cases (users either accurately or inaccurately respond to it). These results are aligned with other studies that considered player type ratings as fixed throughout the experience [4] (see method *A* of our experiments). Nevertheless, previous studies did not consider the dynamic player type method. Since it will not be possible to know whether (real) users accurately answer the player type questionnaire in real situations, our method is the most suitable for an adaptive gamified experience.

Our adaptive method needs information about the player type of the user at the beginning of the experience, so we used the HEXAD player types questionnaire proposed by [20]. It is worth noting that a recent study that describes a new method to predict player types using gameful applications [23]. As we mentioned in Section 3, we selected 14 game elements of the 52 presented in [24]. In the case of extending the proposal from 14 to 52 game elements, our adaptive method is still applicable, it would only be necessary to extend the GE and GM vectors of Equations (3) and (4) and the dimensions of the matrix M , Equation (5). Moreover, our approach is easily applicable to different player types taxonomies such as MoMo [21], a new promising model created as a combination of others, which will be considered in our future research.

However, our system has some limitations. Actually, it is necessary to test different values of updating the frequency of execution of player ratings (Step 1 of the method, as can be seen in Figure 2). Current simulations have used a frequency of one iteration every 15 s, ensuring that changes in the bots' behavior can be detected without overloading the system performance, but this value can be better tuned and even more so when considering real-life scenarios. Moreover, analyzing real users' experiences, we could tune some constants such as ϵ or the values of the matrix M to better adapt the users' profile and the gamified experience. Note that in the process of measuring the error Err in the simulated scenario, the average distance between $PR^{(t)}$ and RPT_0 is considered, and not the distance between $PR^{(t)}$ and RPT_t . However, it would be possible to calculate the error with respect to a dynamic RPT_t , in which case we would slightly randomly change some values of RPT_t , and therefore, from this value, the behavior of the bots (Θ , Υ) would be dynamically calculated. Additionally, a simulation of a bigger change of the real player profile would be possible since it consists of carrying out our simulation n times with a different RPT_0 each time. Indeed, the error is currently calculated as the mean of the distances between $PR^{(t)}$ and RPT_0 of all the types of players (see Equation (12)), which may hide the particular behavior of individual player types (e.g., disruptor, free spirit).

The current evaluation performed using bots should be extended to real users. Thus, additional users' experience data, such as the course and activities completion rate and users' satisfaction, could be used to analyze the effectiveness of the method. The current case study, which is a short course composed of a few activities developed during a short period of time, is somehow limited in terms of the number of collected opinions and interactions. Our method should be applied to other larger gamified systems. Additionally, our method has a limited representation of user engagement behavior using just interactions, interaction speed and opinions. However, considering the more sophisticated variables in the definition of the interaction index, such as eye tracking, cursor tracking or emotion recognition could help us reveal more information about the real player type during the experience.

7. Conclusions

This research proposes a method to present game elements to users that fit their profile (player type). However, instead of taking a (static) picture of the profile at the beginning of the experience, we consider how it may change over the course of gamified activities. The method calculates the utility of showing a game element to the user based on their evolved player type (PT), their interactions with game elements, and the scores given by the users to those game elements. We developed a case study in the educative context, a course integrated on Nanomoocs, a massive open online course (MOOC) platform. We also present a preliminary simulation of the system using bots. A bot simulated three different methods: constant player type ratings, random dynamic player type ratings, and dynamic player type ratings, where the player type is recomputed using players' interactions and opinions at each step of the iterative method. The evaluation shows positive results as the comparative analysis of our proposal yields lower errors when converging towards the real player type for both cases with accurate and inaccurate answers in the player type questionnaire (mean \pm SD with accurate answers–mean \pm SD with inaccurate answers) (0.0070 ± 0.00166 – 0.0243 ± 0.00475) than the static player type (0 ± 0 – 0.0311 ± 0.00404) and the dynamic random player type (0.08024 ± 0.00052 – 0.08027 ± 0.00040). Note that the method of static player type with accurate answers gives 0 error because it is a static profile using accurate values of player types. In this paper, we evaluated the bots' simulations defining the error as a mean of distances between the current and real player type vectors. In future works, we plan to perform an in-depth analysis of the simulation error of each individual component of these vectors. Further work is planned to also include tests with users and incorporate new inputs to our method such as emotions and activity completion.

Author Contributions: The individual contributions for each author are: I.R. and A.P.: conceptualization, investigation, supervision, writing—review and editing, resources; À.R.: conceptualization, investigation, software, data curation, writing—original draft, writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by ACCIO project number COMRDI18-1-0010 and by MCIN/AEI/10.13039/501100011033 grants numbers PGC2018-096212-B-C33 (MISMIS project) and PID2019-104156GB-I00 (Ci-SUSTAIN project).

Acknowledgments: To Accio COMRDI18-1-0010, MISMIS-Language (PGC2018-096212-B-C33) and CI-SUSTAIN (Grant PID2019-104156GB-I00 funded by MCIN/AEI/10.13039/501100011033).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Rapp, A.; Alessandro, M.; Simeoni, R.; Console, L. Playing while Testing: How to Gamify a User Field Evaluation. In *Designing Gamification: Creating Gameful and Playful Experiences Held in Conjunction with SIGCHI Conference on Human Factors in Computing Systems (CHI'13)*; Gamification Research Network: Paris, France, 2013.
2. Cheng, R.; Vassileva, J. Adaptive Reward Mechanism for Sustainable Online Learning Community. In *Proceedings of the Artificial Intelligence in Education, Amsterdam, The Netherlands, 18–22 July 2005*; pp. 152–159.

3. Gené, O.; Núñez, M.; Blanco, A. Gamification in MOOC: Challenges, opportunities and proposals for advancing MOOC model. In Proceedings of the Second International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'14), Salamanca, Spain, 1–3 October 2014; pp. 215–220. [[CrossRef](#)]
4. Lavoue, E.; Monterrat, B.; Desmarais, M.; George, S. Adaptive Gamification for Learning Environments. *IEEE Trans. Learn. Technol.* **2019**, *12*, 16–28. [[CrossRef](#)]
5. Charles, D.; Black, M. Dynamic Player Modelling: A Framework for Player-centred Digital Games. In Proceedings of the 5th International Conference on Computer Games: Artificial Intelligence, Design and Education (CGAIDE'04), Microsoft Academic Campus, Reading, UK, 8–10 November 2004; pp. 29–35.
6. Hallifax, S.; Serna, A.; Marty, J.; Lavoué, G.; Lavoué, E. Factors to consider for tailored gamification. In Proceedings of the CHI PLAY 2019 Annual Symposium on Computer-Human Interaction in Play, Barcelona, Spain, 22–25 October 2019; pp. 559–572. [[CrossRef](#)]
7. Rodríguez, I.; Puig, A.; Rodríguez, A. We Are Not the Same Either Playing: A Proposal for Adaptive Gamification. In *Frontiers in Artificial Intelligence and Applications*; Villaret, M., Alsinet, T., Fernández, C., Valls, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2021; Volume 339, pp. 19–34. [[CrossRef](#)]
8. Hallifax, S.; Serna, A.; Marty, J.; Lavoué, E. Adaptive Gamification in Education: A Literature Review of Current Trends and Developments. *Lect. Notes Comput. Sci.* **2019**, *11722*, 294–307.
9. Klock, A.C.T.; Gasparini, I.; Pimenta, M.S.; Hamari, J. Tailored gamification: A review of literature. *Int. J. Hum.-Comput. Stud.* **2020**, *144*, 102495. [[CrossRef](#)]
10. Chalco, G.C.; Moreira, D.A.; Mizoguchi, R.; Isotani, S. An ontology engineering approach to gamify collaborative learning scenarios. In *CYTED-RITOS International Workshop on Groupware*; Baloian, N., Burstein, F., Ogata, H., Santoro, F., Zurita, G., Eds.; Springer: Berlin/Heidelberg, Germany, 2014; pp. 185–198.
11. Denden, M.; Tlili, A.; Essalmi, F.; Jemni, M. Does personality affect students' perceived preferences for game elements in gamified learning environments? In Proceedings of the IEEE 18th International Conference on Advanced Learning Technologies, ICALT 2018, Mumbai, India, 9–13 July 2018; pp. 111–115. [[CrossRef](#)]
12. Ferro, L.S.; Walz, S.P.; Greuter, S. Towards personalised, gamified systems. In Proceedings of the 9th Australasian Conference on Interactive Entertainment Matters of Life and Death—IE '13, Melbourne, Australia, 30 September–1 October 2013; ACM Press: New York, NY, USA, 2013; pp. 1–6. [[CrossRef](#)]
13. Borges, S.; Mizoguchi, R.; Durelli, V.H.S.; Bittencourt, I.; Isotani, S. A link between worlds: Towards a conceptual framework for bridging player and learner roles in gamified collaborative learning contexts. In *Advances in Social Computing and Digital Education, Croatia*; Koch, F., Koster, A., Primo, T., Guttman, C., Eds.; Springer: Berlin/Heidelberg, Germany, 2016; pp. 19–34.
14. Knutas, A.; Ikonen, J.; Maggiorini, D.; Ripamonti, L.; Porras, J. Creating student interaction profiles for adaptive collaboration gamification design. *Int. J. Hum. Cap. Inf. Technol. Prof.* **2016**, *7*, 47–62. [[CrossRef](#)]
15. Chtouka, E.; Guezguez, W.; Amor, N.B. Reinforcement learning for new adaptive gamified LMS. In Proceedings of the International Conference on Digital Economy, Mumbai, India, 9–13 July 2018; Jallouli, R., Bach Tobji, M., Bélisle, D., Mellouli, S., Abdallah, F., Osman, I., Eds.; Springer: Berlin/Heidelberg, Germany, 2019; pp. 305–314.
16. Monterrat, B.; Lavoué, E.; George, S. Toward an Adaptive Gamification System for Learning Environments. In *Computer Supported Education*; Springer International Publishing: Cham, Switzerland, 2015; pp. 115–129.
17. Lopez, C.; Tucker, C. Toward Personalized Adaptive Gamification: A Machine Learning Model for Predicting Performance. *IEEE Trans. Games* **2020**, *12*, 155–168. [[CrossRef](#)]
18. Bartle, R.A. Hearts, Clubs, Diamonds, Spades: Players Who Suit Muds. *J. MUD Res.* **1996**, *1*, 19.
19. Nacke, L.E.; Bateman, C.; Mandryk, R.L. BrainHex: A neurobiological gamer typology survey. *Entertain. Comput.* **2014**, *5*, 55–62. [[CrossRef](#)]
20. Marczewski, A. Even Ninja Monkeys Like to Play: Gamification, Game Thinking and Motivational Design. In *Game Thinking & Motivational Design*; CreateSpace Independent Publishing Platform, Ed.; Blurb Inc.: London, UK, 2015; pp. 65–80.
21. Sienel, N.; Münster, P.; Zimmermann, G. Player-Type-based Personalization of Gamification in Fitness Apps. In Proceedings of the HEALTHINF, Vienna, Austria, 11–13 February 2021; pp. 361–368.
22. Mora, A.; Tondello, G.F.; Nacke, L.E.; Arnedo-Moreno, J. Effect of personalized gameful design on student engagement. In Proceedings of the IEEE Global Engineering Education Conference, EDUCON, Santa Cruz de Tenerife, Spain, 17–20 April 2018; pp. 1925–1933. [[CrossRef](#)]
23. Altmeyer, M.; Tondello, G.F.; Krüger, A.; Nacke, L.E. HexArcade: Predicting Hexad User Types by Using Gameful Applications. In Proceedings of the CHI PLAY 2020 Annual Symposium on Computer-Human Interaction in Play, Virtual Event, 2–4 November 2020; ACM: New York, NY, USA, 2020; pp. 219–230. [[CrossRef](#)]
24. Tondello, G.F.; Wehbe, R.; Diamond, L.; Busch, M.; Marczewski, A.; Nacke, L.E. The Gamification User Types Hexad Scale. In Proceedings of the 2016—CHI PLAY 2016 Annual Symposium on Computer-Human Interaction in Play, Austin, TX, USA, 16–19 October 2016; pp. 229–243.
25. Hallifax, S.; Lavoué, E.; Serna, A. To tailor or not to tailor gamification? An analysis of the impact of tailored game elements on learners' behaviours and motivation. *Lect. Notes Comput. Sci.* **2020**, *12163*, 216–227. [[CrossRef](#)]
26. Sabourin, J.; Lester, J. Affect and Engagement in Game-Based Learning Environments. *IEEE Trans. Affect. Comput.* **2014**, *5*, 45–56. [[CrossRef](#)]

27. Dalponte Ayastuy, M.; Torres, D.; Fernández, A. Adaptive gamification in Collaborative systems, a systematic mapping study. *Comput. Sci. Rev.* **2021**, *39*, 100333. [[CrossRef](#)]
28. Škuta, P.R.; Kostolányová, K. Adaptive approach to the gamification in education. In Proceedings of the European Conference on Technology Enhanced Learning, Transforming Learning with Meaningful Technologies, Delft, The Netherlands, 16–19 September 2018; Springer International Publishing: Berlin/Heidelberg, Germany, 2018; p. 367.
29. Barata, G.; Gama, S.; Jorge, J.; Gonçalves, D. Gamification for smarter learning: Tales from the trenches. *Smart Learn. Environ.* **2015**, *2*, 1–23. [[CrossRef](#)]
30. Knutas, A.; Van Roy, R.; Hynninen, T.; Granato, M.; Kasurinen, J.; Ikonen, J. A process for designing algorithm-based personalized gamification. *Multimed. Tools Appl.* **2019**, *78*, 13593–13612. [[CrossRef](#)]
31. Hassan, M.A.; Habiba, U.; Majeed, F.; Shoab, M. Adaptive gamification in e-learning based on students' learning styles. *Interact. Learn. Environ.* **2021**, *29*, 545–565. [[CrossRef](#)]
32. Jagušt, T.; Botički, I.; So, H.J. Examining competitive, collaborative and adaptive gamification in young learners' math learning. *Comput. Educ.* **2018**, *125*, 444–457. [[CrossRef](#)]
33. Kickmeier-Rust, M.D. *An Alien's Guide to Multi-Adaptive Educational Computer Games*; Informing Science: Santa Rosa, CA, USA, 2012.
34. Cooper, S.; Khatib, F.; Treuille, A.; Barbero, J.; Lee, J.; Beenen, M.; Leaver-Fay, A.; Baker, D.; Popović, Z. Predicting protein structures with a multiplayer online game. *Nature* **2010**, *466*, 756–760. [[CrossRef](#)] [[PubMed](#)]
35. Eveleigh, A.; Jennett, C.; Lynn, S.; Cox, A.L. "I want to be a captain! I want to be a captain!" gamification in the old weather citizen science project. In Proceedings of the first International Conference on Gameful Design, Research, and Applications, Stratford, ON, Canada, 2–4 October 2013; pp. 79–82.
36. Ismailović, D.; Pagano, D.; Brügge, B. Wemakewords-An adaptive and collaborative serious game for literacy acquisition. In Proceedings of the IADIS International Conference-Game and Entertainment, Rome, Italy, 20–26 July 2011.
37. Cantwell, D.; Broin, D.O.; Palmer, R.; Doyle, G. Motivating elderly people to exercise using a social collaborative exergame with adaptive difficulty. In Proceedings of the 6th European Conference on Games Based Learning, Cork, Ireland, 4–5 October 2012; pp. 4–5.
38. Llanos, J.; Carro, R.M. The squares: A multi-touch adaptive game for children integration. In Proceedings of the 2015 International Symposium on Computers in Education (SIIE), Setubal, Portugal, 25–27 November 2015; pp. 137–140.
39. Tranvouez, E.; Fournier, S. A MultiAgent architecture for collaborative serious game applied to crisis management training: Improving adaptability of non player characters. *EAI Endorsed Trans. Serious Games* **2014**, *1*, e7. [[CrossRef](#)]
40. Knutas, A.; van Roy, R.; Hynninen, T.; Granato, M.; Kasurinen, J.; Ikonen, J. Profile-Based Algorithm for Personalized Gamification in Computer-Supported Collaborative Learning Environments. In Proceedings of the 1st Games-Humans Interactions GHITALY@CHIItaly, Cagliari, Italy, 18 April 2017.
41. Yu, H.; Riedl, M.O. A sequential recommendation approach for interactive personalized story generation. In Proceedings of the AAMAS, Valencia, Spain, 4–8 June 2012; Volume 12, pp. 71–78.
42. Hastings, E.J.; Guha, R.K.; Stanley, K.O. Automatic content generation in the galactic arms race video game. *IEEE Trans. Comput. Intell. AI Games* **2009**, *1*, 245–263. [[CrossRef](#)]
43. Hassan, M.A.; Habiba, U.; Khalid, H.; Shoab, M.; Arshad, S. An adaptive feedback system to improve student performance based on collaborative behavior. *IEEE Access* **2019**, *7*, 107171–107178. [[CrossRef](#)]
44. Feyisetan, O.; Simperl, E.; Van Kleek, M.; Shadbolt, N. Improving paid microtasks through gamification and adaptive furtherance incentives. In Proceedings of the 24th International Conference on World Wide Web, Florence, Italy, 18–22 May 2015; pp. 333–343.
45. Mora, A.; González, C.; Arnedo-Moreno, J.; Álvarez, A. Gamification of cognitive training: A crowdsourcing-inspired approach for older adults. In Proceedings of the XVII International Conference on Human Computer Interaction, Salamanca, Spain, 13–16 September 2016; pp. 1–8.
46. Emmerich, K.; Neuwald, K.; Othlinghaus, J.; Ziebarth, S.; Hoppe, H.U. Training conflict management in a collaborative virtual environment. In Proceedings of the International Conference on Collaboration and Technology, Raesfeld, Germany, 16–19 September 2012; Herskovic, V., Hoppe, H.U., Jansen, M., Ziegler, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 17–32.