Challenge Balancing for Personalised Game Spaces

Sander Bakkes, Shimon Whiteson, Guangliang Li,
George Viorel Vişniuc, Efstathios Charitos, Norbert Heijne, and Arjen Swellengrebel
University of Amsterdam, Intelligent Systems Laboratory, Amsterdam, The Netherlands
Email: {s.c.j.bakkes, s.a.whiteson, g.li}@uva.nl
{george.visniuc, efstathios.charitos, norbert.heijne, arjen.swellengrebel}@student.uva.nl

***Abstract*—In this paper we propose an approach for personalising the space in which a game is played (i.e., levels) – to the end of tailoring the experienced challenge to the individual user *during actual play* of the game. Our approach specifically considers two design challenges, namely *implicit user feedback* and *high risk of user abandonment*. We contribute an approach that acknowledges that for effective online game personalisation, one needs to (1) offline learn a policy that is appropriate in expectation across users – to be used for initialising the online game, (2) offline learn a mapping from gameplay observations to the player experience – to be used for guiding the online game personalisation, and (3) rapidly converge to an appropriate policy for the individual user in online gameplay – employing the learned feedback model and a straightforward model of user abandonment. User studies that validate the approach to online game personalisation in the actual video game INFINITE MARIO BROS. indicate that it provides an effective basis for automatically balancing the game's challenge level to the individual human player.**

I. Introduction

Ideally, artificial intelligence (AI) in games provides satisfactory and effective game experiences for players regardless of gender, age, capabilities, or experience [1]; it allows for the creation of personalised games, where the game experience is continuously tailored to fit the individual player. Indeed, we are now at a point where modern computer technology, simulation, and artificial intelligence (AI) have opened up the possibility that more can be done with regard to *on-demand* and *just-in-time personalisation* [2]. As such, achieving the ambition of creating personalised games requires the development of novel techniques for assessing online and unobtrusively which game adaptations are required for optimising the individual player's experience.

The goal of the present research is to online generate game spaces (i.e. levels) such that the spaces optimise player challenge for the individual player. This entails two challenges, namely (1) in online gameplay only implicit feedback on the appropriateness of the personalisation actions is available (i.e., the AI can only observe the player interacting with the game, while not being provided with labels on the player experience), and (2) there is a high risk of user abandonment when inappropriate game personalisation actions are performed.

Our contribution is an approach built on three features that address these challenges. Addressing the first challenge, a *feedback model* needs to be learned which maps gameplay observations to an estimate of the player experience. The feedback model is learned offline and is subsequently employed for decision-making in online gameplay. Addressing the second challenge, a policy needs to be learned that is appropriate in expectation across users. This policy is learned offline and is subsequently employed to *initialise online gameplay*. Also addressing the second challenge, in online gameplay we *rapidly converge* to an appropriate policy for the individual user. To do this well, we employ the learned feedback model and a straightforward model of user abandonment to *guide state-space exploration*.

II. Game Personalisation

Game personalisation is motivated by a significantly increased involvement and extensive cognitive elaboration when subjects are exposed to content of personal relevance [3]; they will exhibit stronger emotional reactions [4]. Particularly, a positive effect on player satisfaction is indicated, i.e., game personalisation raises player loyalty and enjoyment, which in turn can steer the gaming experience towards a (commercial) success [5]. Indeed, the perspective of AI researchers to increase the engagement and enjoyment of the player is one that is consistent with the perspective of game designers [2], i.e., personalisation methods are regarded as instrumental for achieving industry ambitions [6]. Tailoring the game experience to the individual player particularly benefits from the use of player models, and requires components that use these models to adapt part of the game [7].

Our research follows the emerging trend of employing AI methods for adapting the game environment itself (as opposed to, more typically, adapting the behaviour of the game characters) [7]. In our investigation, we choose to focus on *personalising the game space* to the individual player with respect to experienced challenge[1]; it can be considered an instantiation of experience-driven procedural content generation (EDPCG) [8]. Related work with regard to this scope is discussed next.

A. Challenge balancing

Challenge balancing concerns automatically adapting the challenge that a game poses to the skills of a player [9], [10]. It aims at achieving a 'balanced game', i.e., a game wherein the player is neither challenged too little, nor challenged too much. In most games, the only implemented means of challenge balancing is provided by a *difficulty setting*, i.e., a discrete parameter that determines how challenging the game will be. However, as the challenge provided by a game is typically *multi-faceted*, it is hard for the player to estimate reliably the challenge level that is appropriate for her. Furthermore, generally only a limited set of discrete difficulty settings is available (e.g., easy, normal, hard). This entails that the

[1]Typically, in a video game setting, the game space will be the levels in which the game takes place. Game space personalisation, in this context, generally concerns the online learning of the next set of level design parameters.

available settings are not fine-tuned to be appropriate for each player. As such, researchers have developed advanced techniques for balancing the challenge level of games. Hunicke and Chapman [11] explored challenge balancing by controlling the strength of opponent characters (i.e., controlling the opponent character's health, accuracy, and employed weapons). Spronck *et al.* [10] investigated methods for automatically adjusting weights assigned to possible game scripts. Zook and Riedl [12] investigated a temporal data-driven player model for dynamic difficulty adjustment. Indeed, knowledge on the specific effect of game adaptations can be employed for maintaining a challenge level [13], and may be incorporated to steer the procedural generation of game content [14].

In our research, we take the distinct focus of balancing the game's challenge level by adapting the *content* that is placed within the game environment.

B. Player modelling

Player modelling is of increasing importance in modern video games [15]; it is almost a necessity when the purpose of AI is 'entertaining the human player' rather than 'defeating the human player' [16]. A challenge for player modelling in video games is that models of the player have to be established (1) in game environments that generally are relatively complex, (2) with typically little time for observation, and (3) often with only partial observability of the environment [17], [18]. A recent development with regard to player modelling is to automatically establish psychologically or sociologically verified player profiles [18]. Such models provide motives or explanations for observed behaviour. A solid profile can be used to, for instance, predict a player's affective state during play of the game.

In our research, we take the distinct approach of utilising player models for the purpose of optimising the player experience (*cf.* the work of Yannakakis *et al.* [19]).

C. Space adaptation

Game space adaptation concerns allowing the space in which the game is played to adapt, ideally in response to the user experience [20]. Game space adaptation is an active area of research [14], [21], [22], [20], [23], which falls within the scope of experience-driven procedural content generation [8]. Research is increasingly focussing on how procedural techniques may be employed specifically for enhancing the player experience. Indeed, the feasibility of procedurally generating a personalised race track has already been demonstrated by Togelius *et al*. [24], [25]. Furthermore, Adrian *et al.* [26] and Traichioiu *et al.* [27] investigated the procedural generation of levels with respect to a difficulty curve. Dormans and Bakkes [20], [28] investigated the procedural generation of entire levels for action-adventure games, which distinguishes between missions and spaces as two separate structures that need to be generated in two individual steps. Lopes *et al.* [29] developed a framework aimed at creating personalised content for complex and immersive game worlds; by modelling which content provided the context for a given personal gameplay experience. Sorenson *et al.* [30] developed an approach to automatic video game level design consisting of a computational model of player enjoyment and a generative system based on evolutionary computing.

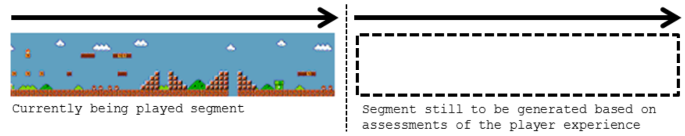


Fig. 1: Our enhanced version of INFINITE MARIO BROS. During gameplay it generates short *new level segments* of specific content *on-the-fly*, on the basis of an implicit reward signal (given by an offline learned feedback model).

In our research, we also focus on procedural content generation for tailoring the player experience. Our distinct focus in this matter, is to assess *online* and *unobtrusively* which game adaptations are required for optimising the individual player's experience while the game is being played, so as to have assessments on the experienced player challenge impact the procedural process.

III. DOMAIN DESCRIPTION

We consider a typical video game: INFINITE MARIO BROS. [31]; an open-source clone of the classic video game SUPER MARIO BROS. It can be regarded an archetypal platform game; despite its relatively straightforward appearance it provides a diverse and challenging gameplay experience. We build upon a version of INFINITE MARIO BROS. that has been extended by Shaker *et al.* [32], [33], [34], Pedersen *et al.* [22], [35], and Togelius *et al.* [36] to procedurally generate Mario levels.

We have made two further enhancements to the 2011 Mario AI Championship game engine of INFINITE MARIO BROS. First, we enhanced the engine such that it is able to procedurally generate *segments* of Mario levels while the game is in progress (Figure 1). One game segment has a width of 112 game objects, and generally takes a player approximately 20 to 30 seconds to complete. This enhancement enables feedback on the observed player experience to rapidly impact the procedural process that generates the upcoming level segments. The upcoming level segments are generated seamlessly, such that no screen tears occur when the user is transitioning from one segment to the next (i.e., before the next segment can be observed a short 'gap' block is injected in the game space).

Our second enhancement to the game engine, is that within every segment we can now inject short *chunks* of specific game content. We enabled the game engine to generate five different types of chunks, (1) a straight chunk, containing enemies and jumpable blocks, (2) a hill chunk, also containing enemies, (3) a chunk with tubes, containing enemy plants, (4) a jump, and (5) a chunk with cannons. Each chunk can have six distinct implementations, stemming from a per-chunk parameter value $\in [0, 5]$. The intended challenge level of a chunk monotonically increases with the parameter value (e.g., a hill parameter value of 0 entails a chunk with no hills and no enemies, while a value of 5 entails five procedurally-generated hills with five relatively difficult enemies). Our enhanced engine has the desired property that the generated chunks are largely independent of each other, i.e., only in rare cases will one chunk be able to affect player behaviour in the surrounding chunks (e.g., in case a cannon bullet follows the player to the next chunk). To benefit playability and level aesthetics, the order in which the chunks are encountered is randomized for each new segment. The five chunks (each 16 game objects in length) are preceded and succeeded by a flat, neutral chunk (also 16 game objects in length), to allow the player to prepare for the next game segment.

In online gameplay, the AI that personalises the game space is input with a vector of 65 real-numbered features values of observed player behaviour. The features encompass the full logging capability of the game's data recorder (45 features, such as *timesjumped*, *coinscollected*, *kills-red-turtle*), appended by 15 hand-coded features (such as *jump-totalruntime*, *cannons-totallefttime*) and 5 parameters (such as *parameter-cannons*).[2] The only action that the AI can take is to output a vector of five integers (chunk parameters) $\in [0, 5]$ to the procedural process which in turn generates the next level segment. While the action space is relatively modest in size, its resulting expressiveness ranges from overly easy to exasperatedly hard level segments.

IV. Approach

The goal of the present research is to online generate game spaces (i.e. levels) such that the spaces optimise player challenge for the individual player, without interrupting the game experience for gathering player feedback. This entails two challenges, namely (1) in online gameplay only implicit feedback on the appropriateness of the personalisation actions is available (i.e., the AI can only observe the player interacting with the game, while not being provided with labels on the player experience), and (2) there is a high risk of user abandonment when inappropriate game personalisation actions are performed.

Our contribution is an approach built on three features that address these challenges. Addressing the first challenge, a *feedback model* needs to be learned which maps gameplay observations to an estimate of the player experience. The feedback model is learned offline and is subsequently employed for decision-making in online gameplay. Addressing the second challenge, a policy needs to be learned that is appropriate in expectation across users. This policy is learned offline and is subsequently employed to *initialise online gameplay*. Also addressing the second challenge, in online gameplay we *rapidly converge* to an appropriate policy for the individual user. To do this well, we employ the learned feedback model and a straightforward model of user abandonment to *guide state-space exploration*. We now describe these ideas in more detail.

A. Phase 1 – Learn the global safe policy (offline)

The procedure adopted for learning the global safe policy, while intelligently generating a set of training instances, is illustrated in Figure 2a; it concerns labelling sessions with human participants. O indicates a gameplay observation (a vector of 65 real-numbered features), P indicates the parameters used for generating a level segment (a vector of five integers $\in [0, 5]$), L indicates the user-provided label of the gameplay experience; it concerns a 5-point Likert scale, being 1=Too easy, 2=Somewhat easy, 3=Just right, 4=Somewhat hard, and 5=Too hard. Each labelling session starts by providing the participant with a 'tutorial' segment of the easiest possible challenge level – so that the participant may get accustomed to interacting with the Mario game and the game controls. Upon completing the tutorial segment, the second segment is generated with a challenge level that is uniformly increased across all parameters by one step size. Upon completing the second segment, we query the participant on two types of feedback, namely (1) the experienced challenge level, on a 5-point Likert scale, and (2) her preference for the segment that she just played, or for the segment before. Depending on the answer on this second type of feedback, a heuristically-determined exploration function is employed for regulating which point in the parameter space should be explored next (the first type of feedback is employed for learning the feedback model, not for exploration at training time).[3] Given the selected point in the parameter space, the next segment is generated and presented to the participant, and the process repeats. For each labelling session, labels are collected till the user chooses to abandon the game.

Algorithm 1: Exploration policy for experience labelling.

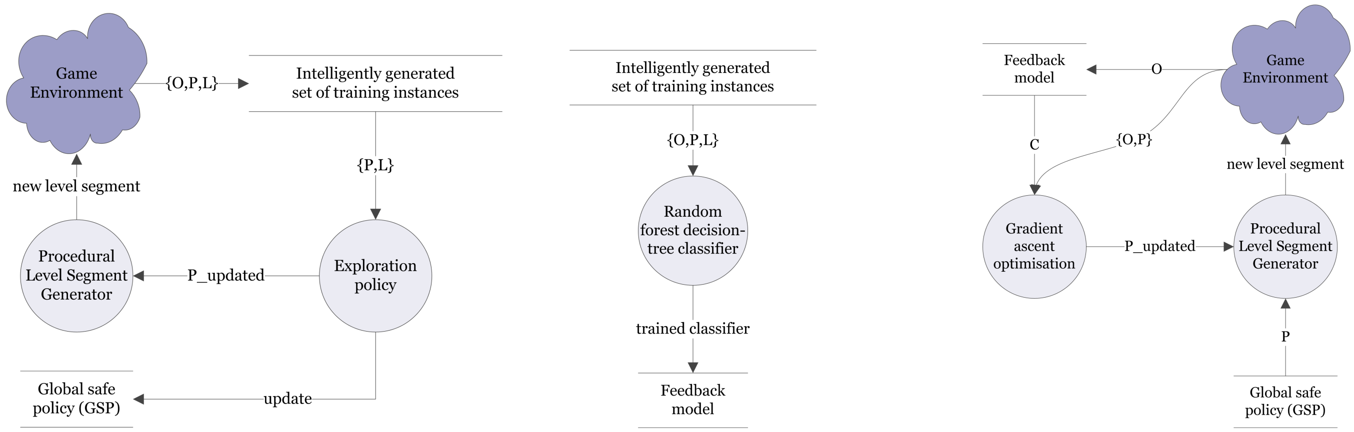
```
i = number-of-controllable-features; j = range-of-feature-values;
fine-tuning = false; GSP[] = p[] = 0; generateNextSegment(p[]);
while playing do
    if player-at-procedural-checkpoint then
        if !fine-tuning then
            if player-prefers-current-segment then
                GSP[] = p[]; inc(p[]);
            else
                fine-tuning = true; p[] = GSP[]; p[rand(0,i-1)] = rand(0,j);
            end
        else
            if player-prefers-current-segment then
                GSP[] = p[];
            else
                p[] = GSP[];
            end
            p[rand(0,i-1)] = rand(0,j);
        end
        generateNextSegment(p[]);
    end
    if player-dies then
        i = feature-associated-to-chunk-of-death;
        p[i] = max(p[i] - rand(0,5), 0); generateNextSegment(p[]);
    end
end
```

Given that human-provided labels are a scarce resource, the exploration policy (Algorithm 1) is aimed at (a) ensuring that the earlier provided labels are as informative as possible, (b) ensuring that the participant will keep playing the game for as long as possible, and (c) enabling an anytime solution, which – in case the participant suddenly abandons the game – at any time can return a reasonable estimate on the participant's preferred challenge level. The algorithm attempts to rapidly assess a challenge level that is roughly appropriate to the concerning participant, and optimise the assessment by exploring around this rough estimate. In case the participant inadvertently dies, the policy decreases the challenge level on specifically the aspect that caused the participant to die (e.g., an excess of cannons in the game space), such that the participant does not get stuck in the game and will continue labelling.

The result of each labelling session is an estimate on the participant's preferred challenge level. Given that (1) the features that express the challenge level are largely independent of each other, and that (2) the intended challenge level

[2]A complete description of the employed features is provided on http://ieva.landofsand.com/not-from-ieva/description-of-features.pdf

[3]This links to advanced techniques for active player modelling [37].



(a) Phase 1: Learn the global safe policy (offline) (b) Phase 2: Learn the feedback model (offline) (c) Phase 3: Online personalisation (online)

Fig. 2: Approach to game-space personalisation (legenda explained in the text).

monotonically increases with the feature value, we consider the global safe policy to be the statistical mean over the preferred challenge levels.

B. Phase 2 – Learn the feedback model (offline)

The procedure adopted for learning the feedback model is illustrated in Figure 2b. Given the intelligently generated set of training instances (Phase 1), a random forest decision-tree classifier [38] is built. The training instances concern tuples of gameplay observations O, the procedural parameters P that were used to generate the observed level segment, and the labels L of the user's experience when playing the segment. We refer to the built random forest decision-tree classifier as a *feedback model*. The random forest decision-tree classifier consists of a combination of tree classifiers where each classifier is generated using a random vector sampled independently from the input vector, and each tree casts a unit vote for the most popular class to classify an input vector [39]. The advantage of using random forest classification is that it returns a probability distribution of the classification (and not solely a, less informative, single classification).

The feedback model returns a classification C, being a probability distribution of the newly observed instance resulting from the five Likert-scale classes of challenge. We employ the classification C for two purposes (during online personalisation), namely (1) to calculate the expected reward, and (2) to calculate the expected Likert-class of experienced challenge. Given the classification C the expectation of the reward R is defined as:

$$E[R] = \sum_{k=1}^{5} r_k p_k, \tag{1}$$

where r_k is the heuristically-determined reward value for Likert-challenge class k, and p_k is the probability of observed player behaviour resulting from challenge class k. We have heuristically determined the following reward values: a challenge label of 3 yields a reward r_k of 1.0, label 2 and 4 yield a reward of 0.33, and label 1 and 5 yield a reward of 0.0. Given the classification C the expectation of the Likert class K is defined as:

$$E[K] = \sum_{k=1}^{5} k p_k, \tag{2}$$

where k is the Likert-challenge class, and p_k is the probability of observed player behaviour resulting from Likert class k. Both Equation 1 and 2 are used for online game personalisation.

C. Phase 3 – Online personalisation (online)

In online play of the game, we initialise the game with a challenge level that is most appropriate in expectation across users (given the global safe policy learned in Phase 1). Subsequently, during play of the game we employ the learned feedback model (learned in Phase 2) to guide the search through the state space. A naive approach to this end would be to perform gradient-ascent optimisation, using the classification of the observed player behaviour to determine the search direction and step-size.[4] We enhance the naive gradient-ascent optimisation approach by incorporating a basic, straightforward model of user abandonment. Indeed, such a model may effectively guide the online state-space exploration; game content that is associated with a high probability of user abandonment should be adapted differently from game content that is associated with a low probability of user abandonment. Our abandonment model follows the intuition that the risk that a player takes when playing the game is proportionally related to the probability of abandonment. Our assumption is that when a player exhibits risky behaviour in specific Mario chunks (e.g., the player is continuously running in the cannons section), she may be doing so because the chunk is not sufficiently challenging (and vice versa). Consequently, for the Mario domain, we consider the fraction of time that a player spent running in the concerning chunk as a rough estimate of the probability of user abandonment.

The procedure adopted for online personalisation is illustrated in Figure 2c. First, at the start of a game, the global safe policy GSP (learned in Phase 1) is fed into the parameters P that are employed for generating the first level segment

[4] Were the implicit feedback reliable enough to be interpreted as supervised labels, it could be employed to directly jump to the predicted best parameter values. Instead, the feedback provides relative information (basically an implicit reward signal), which can be used to determine a gradient to follow. This is a safer approach, since it leads to incremental changes that accumulate over time, rather than a big jump based only on a single data point.

ALGORITHM 2: Online personalisation.

```
i = number-of-controllable-features; j = range-of-feature-values; p[] = GSP[];
generateNextSegment(p[]);
while playing do
    if player-at-procedural-checkpoint then
        E[R] = cf. Equation 1; E[K] = cf. Equation 2;
        stepSize = α · maxStep · (1 - E[R]);
        for n=0; n<i; n++ do
            if E[K] <= desired-Likert-class then
                p[n] = p[n] + (probUserAbandonment[n] · stepSize);
            else
                p[n] = p[n] - ((1-probUserAbandonment[n]) · stepSize);
            end
        end
        generateNextSegment(p[]);
    end
    if player-dies then
        i = feature-associated-to-chunk-of-death;
        p[i] = max(p[i] - 1, 0); updateCurrentSegment(p[]);
    end
end
```

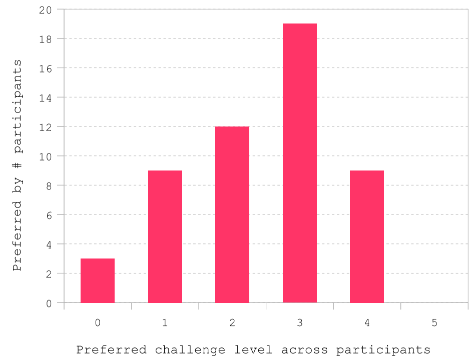


Fig. 3: Histogram of the preferred challenge level across users. The bins reflect the average attribute values over all dimensions.

(it is appropriate in expectation across users). Just before completion of a level segment by the player (i.e., after having completed a segment length of 96 out of the total length of 112), gameplay observations O are fed into the feedback model (learned in Phase 2). The resulting classification C by the feedback model – a probability distribution over the five Likert classes, as returned by the learned feedback model – is input to the gradient-ascent optimisation method, together with the actual gameplay observations O and the parameters P that were used to generate the level segment. In the gradient-ascent method, the parameters P are mutated in a direction that is more appropriate in expectation to the challenge experienced by the player. The resulting updated P is fed into the procedural level segment generator. With a new level segment being generated, the process now repeats. In some additional detail (*cf.* Algorithm 2); at every procedural checkpoint, given the classification C, the expected reward E[R] and expected Likert class E[K] are estimated by Equation 1 and 2, respectively. The E[R] estimate is used to determine the step size for the gradient ascent optimisation. The E[K] estimate is used to determine the search direction; it increases (decreases) the challenge level with an E[K] below or equal (above) the desired level. The actual step taken in this direction *per procedural parameter*, is weighted according to the estimated probability of user abandonment.

V. EXPERIMENTS

Here we discuss the experiments that validate our approach in the actual video game INFINITE MARIO BROS.

A. Learning the global safe policy (Phase 1)

Following the procedure described in Section IV, we gathered training instances from 52 unique user sessions; the employed parameters were i=5 and j=5. Each session yielded the preferred challenge policy per participant. Given the described procedure, the global safe policy across users was calculated to be: straights=2, hills=3, tubes=3, jumps=2, and cannons=2. This global safe policy concerns a moderately easy challenge level.

The learned global safe policy is strictly data driven, and reflects the statistical mean of preferred challenge levels across users. Further analysis (Figure 3) reveals that the majority of human participants prefer an easy to moderately easy challenge level; while few to no participants prefer a hard to very hard challenge level, respectively. These observations are in line with our own expertise with the problem domain.

B. Learning the feedback model (Phase 2)

While exploring the parameter space for the purpose of learning a global safe policy, we have *de facto* explored that part of the parameter space that is most informative for learning a feedback model (i.e., the part of the parameter space that is not covered by domain knowledge of the game designer). The so gathered training instances are used for learning the feedback model; they are used for building a random forest decision-tree classifier [38]. The classifier is built from 435 labelled instances, generated from 52 unique user sessions.

Utilising 10-fold cross validation, over the five distinct classes of experienced challenge, the random forest classifier achieves an accuracy of 54.02%. In comparison, a random classification yields an accuracy of 20%, and a Naive Bayes-, J48-, and RBF network classifier yield an accuracy of 38.16%, 40.92%, and 46.66%, respectively. As such, we consider the learned feedback model comparatively accurate, particularly as in our experiments we are interested in the *skew of the probability distribution* that is returned by the random forest classifier (as opposed to being concerned with the classification of a single class). That is, the skew of the probability distribution determines the search direction of the gradient-ascent optimisation in online gameplay. The skew accuracy of the learned feedback model is 95.40% (the baseline of selecting a random direction having an accuracy of 50%), with 2.07% of the instances being incorrectly classified as over-challenging, and 2.53% of the instances being incorrectly classified as under-challenging.

We examined which features are of most merit (i.e., concerning information gain) in classifying the experienced player challenge, via the ReliefF attribute evaluation method [40]; a robust and noise-tolerant algorithm for attribute evaluation [41]. The features with the most merit are given in Table I. We observe that the majority of these features concern (1)

TABLE I: Top ranking features according to the ReliefF atribution evaluation method. Controllable features are highlighted in bold-face.

Average merit	Feature
0.032 $\pm$ 0.004	**parameter-hills**
0.027 $\pm$ 0.003	**parameter-jump**
0.021 $\pm$ 0.003	**parameter-cannons**
0.020 $\pm$ 0.002	**parameter-tubes**
0.015 $\pm$ 0.002	feature-tubes-totalruntime
0.013 $\pm$ 0.002	feature26-aimlessJumps
0.013 $\pm$ 0.002	**parameter-straight**
0.012 $\pm$ 0.002	feature21-emenies
0.011 $\pm$ 0.002	feature-straight-totalruntime
0.010 $\pm$ 0.001	feature-straight-totallefttime

controllable features (highlighted in bold-face), and (2) the time that a participant has spent running through part of the level. These latter features reflect the exhibited risk that a participant is taking while playing the game. These two findings support our intuition that (a) the adopted controllable features (i.e., the parameters of the generative process) may well be employed for controlling the challenge level, and that (b) at least in the Mario game, indicators on exhibited player risk effectively contribute to the classification of the experienced player challenge.

C. Online personalisation (Phase 3) – Performance analysis

In this experiment, we analyse the system's performance by observing human participants interact with the gaming system under four experimental conditions, namely (**1a**) starting at a random challenge level + maintaining that level, (**1b**) starting at a random challenge level + game personalisation, (**1c**) starting at the global safe policy + maintaining the policy, and (**1d**) starting at the global safe policy + game personalisation. Our hypothesis is that in condition 1b the experienced challenge level will converge to the desired level (whereas in 1a it will plateau at an undesired level), and that in condition 1d the experience challenge will converge more rapidly (whereas in condition 1c it will plateau as well; albeit at a more desired level than in condition 1a).

The performance of the system is evaluated in terms of the system's ability to converge to – and maintain – a state that is considered appropriate to the human participants (i.e., an adequately balanced challenge level of '3').[5] We employed the learned global safe policy (Section V-A), the learned feedback model (Section V-B), and the parameters i=5, j=5, GSP={2,3,3,2,2}, α=0.8, and maxStep=5.

The experiment is performed by twenty-five humans participants. To minimise user fatigue impacting the experimental results, each of the four game-playing session is ended after a maximum of 10 level segments (i.e., approximately seven minutes of play). An experiment coordinator observes the human participant play the game, and at the end of every segment denotes the experienced challenge – as vocalised by the participant as the applicable Likert-scale class ranging from 1 to 5. Should the participant decide to abandon the session, then this is denoted as 'DNF'. The participant demographics for this experiments were, gender: 24% female, 76% male,

[5]Depending on the context, one may prefer to converge to a challenge level that is, e.g., slightly above what is deemed appropriate to the individual user.

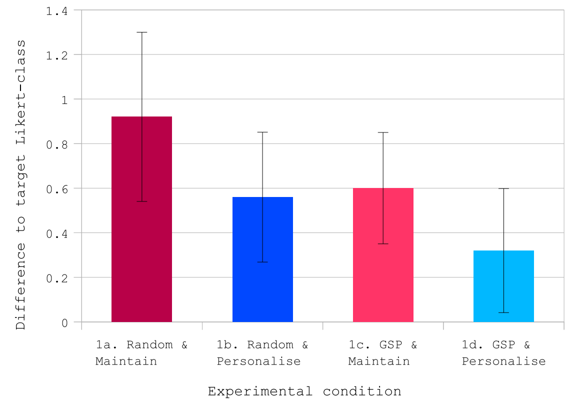


Fig. 4: Online personalisation – Learning performance after the first segment.

age: 27 years (sd=4), hours spent on video games per week: 6 hours (sd=8), has played Mario before: yes for all participants.

Figure 4 illustrates the measured difference to the target Likert-class after a human participant has completed the first segment. Our target Likert-class is 3, so the maximum obtainable difference is 2. We observe that for condition 1a and 1b, the difference to the target Likert-class averaged 0.92 (sd=0.76) and 0.56 (sd=0.58), respectively. This substantial difference (p=0.066) suggests that when online personalisation is enabled (condition 1b), our approach can rapidly learn an appropriate policy: already before the first segment is completed by the human participant. Indeed, it reveals the importance of learning from specific gameplay events already early in the game (i.e., player deaths in the Mario game). Furthermore, we observe that for condition 1c and 1d, the difference to the target Likert-class averaged 0.6 (sd=0.5) and 0.32 (sd=0.56), respectively. This substantial difference (p=0.068) suggests that starting at an appropriate global safe policy (condition 1d), effectively assists the online learning process.

While the variance in the results is large due to the low number of participants, the personalisation system does appear to learn fast and effectively at the beginning of the game, so that further improvements may not be necessary or possible. What we desire from the system at this point, is to stably maintain what it has learned. Figure 5 illustrates the performance of the approach over the course of 10 level segments, after appropriate behaviour has been learned in the first segment (condition 1b). The data points are averaged over 25 game sessions, with missing data points resulting from abandoned sessions being excluded from the average. The figure suggests that the personalisation approach is indeed able to maintain the learned policy in the face of behavioural noise.

Table II lists the number of sessions that were abandoned by the human participants. We observe that when online game personalisation was *not* employed (condition 1a + 1c), 6 out of 50 sessions were abandoned by the human participants. In contrast, we observe that when online game personalisation was employed (condition 1b & 1d), only 1 out of 50 sessions were abandoned by the human participant. This result suggests that, in the face of user abandonment, online game personali-

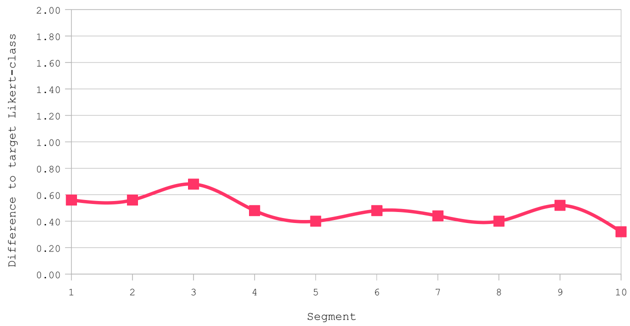


Fig. 5: Relative stability of the online personalisation approach after appropriate behaviour has been learned in the first segment (condition 1b).

TABLE II: Online personalisation – Sessions abandoned.

Experimental condition	Sessions started	Sessions abandoned
1a. Random & Maintain	25	6
1b. Random & Personalise	25	0
1c. GSP & Maintain	25	0
1d. GSP & Personalise	25	1

sation is an effective method for recovering from inappropriate starting conditions.

D. Online personalisation (Phase 3) – Pairwise tests

In this experiment, we investigate how human participants experience the personalised game under actual game playing conditions, in comparison with a realistic (baseline) static game. To this end, in accordance with procedures employed by Shaker *et al.* [34], we query for *pairwise preferences* (i.e., "is system A preferred over system B?"), a methodology with numerous advantages over rating-based questionnaires (e.g., no significant *order of reporting* effects) [42]. We perform pairwise tests of a static system s, with a fixed difficulty level, and a personalised system p. The experiment follows a *within-subjects design* composed of two randomised conditions (first s then p, or inversely), each condition consisting of a series of three sequentially performed pairwise tests, in *randomized order* (i.e., both the condition is randomised, and the order of each pair-wise test is randomised). The pairwise tests compare (**2a**) the static system vs. the personalised system, both starting at identical challenge levels (i.e., disabled smart cold-start initialisation), (**2b**) the static system vs. the personalised system with enabled smart cold-start initialisation (i.e., starting at the learned global safe policy), and (**2c**) the personalised system with disabled smart cold-start initialisation vs. the personalisation system with enabled smart cold-start initialisation. Table III gives an overview of the resulting experimental conditions, with the initial challenge level of a system indicated between brackets.

The experiment is performed by fifteen human participants. To minimise user fatigue impacting the experimental results, each of the nine game-playing session is ended after a maximum of 3 level segments (i.e., approximately two minutes of play). After completing a pair of two games, we query the participants's preference through a 4-alternative forced choice (4-AFC) questionnaire protocol (e.g., s is preferred to p, p is

TABLE III: Experiment 2 – Pairwise tests of the static system versus the personalised system. The initial challenge level is indicated between brackets.

Experiment	Condition 1	Condition 2
2a	s(easy) vs. p(easy) s(normal) vs. p(normal) s(hard) vs. p(hard)	p(easy) vs. s(easy) p(normal) vs. s(normal) p(hard) vs. s(hard)
2b	s(easy) vs. p(gsp) s(normal) vs. p(gsp) s(hard) vs. p(gsp)	p(gsp) vs. s(easy) p(gsp) vs. s(normal) p(gsp) vs. s(hard)
2c	p(easy) vs. p(gsp) p(normal) vs. p(gsp) p(hard) vs. p(gsp)	p(gsp) vs. p(easy) p(gsp) vs. p(normal) p(gsp) vs. p(hard)

TABLE IV: Online personalisation – Pairwise preferences of participants.

Experiment	Pers.(gsp)	Pers.(fixed)	Static	Both	Neither
2a	–	62.22%	22.22%	11.11%	4.44%
2b	68.89%	–	20.00%	6.67%	4.44%
2c	64.44%	28.89%	–	2.22%	4.44%

preferred to s, both are preferred equally, neither is preferred; both are equally unpreferred). The question presented to the participant is: "For which game did you find the challenge level more appropriate?". The participant demographics for this experiments were, gender: 27% female, 73% male, age: 27 years (sd=5), hours spent on video games per week: 5 hours (sd=6), has played Mario before: yes for all participants. We employ the same feedback model, global safe policy, and parameter settings as in Experiment 1. The challenge levels easy, normal, and hard reflect parameter values of {1,1,1,1,1}, {3,3,3,3,3}, and {5,5,5,5,5}, respectively.

Table IV lists the pairwise preferences as reported by the human participants. We observe that, in condition 2a, a significant majority (p=0.0206) of human participants prefer the personalised system over the static system (62.22% over 22.22%) when both systems are initialised with the same policy. In addition, in condition 2b, we observe that when the personalised system is initialised with the learned global safe policy, the significant preference for this system increases to 68.89% (while the preference for the static system remains low, at 20.00%). Finally, in condition 2c, we observe that the personalised system – initialised with the learned global safe policy – is significantly preferred over the personalised system with fixed (easy/normal/hard) initialisations (p=0.0456), 64.44% over 28.89%, respectively.

From these results we may conclude that (a) human participants consistently prefer the personalised gaming system over the static gaming system, and (b) learning an appropriate global safe policy for initialising the game positively affects participant preferences.

VI. Conclusion

In this paper we proposed an approach for personalising the space in which a game is played (i.e., levels) – to the end of tailoring the experienced challenge to the individual user *during actual play* of the game. Our approach specifically considers two design challenges, namely *implicit user feedback* and *high risk of user abandonment*. We contributed an approach that acknowledges that for effective online game personalisation, one needs to (1) offline learn a policy that is appropriate in expectation across users – to be used for initialising the online game, (2) offline learn a mapping from gameplay observations to the player experience – to be used for guiding the online game personalisation, and (3) rapidly converge to an appropriate policy for the individual user in online gameplay – employing the learned feedback model and a straightforward model of user abandonment.

Conclusion. User studies that validated the approach to online game personalisation in the actual video game INFINITE MARIO BROS. indicate that (a) the approach can rapidly learn an appropriate policy: already before the first segment is completed by the human participant, (b) starting at an appropriate global safe policy effectively assists the online learning process, (c) the approach can maintain the learned policy in the face of behavioural noise, (d) in the context of user abandonment, online game personalisation is an effective method for recovering from inappropriate starting conditions, (e) in pairwise tests, a significant majority of human participants prefer the personalised gaming system over a static gaming system, and (f) learning an appropriate global safe policy for initialising the game positively affects participant preferences. From these results, we may conclude that the proposed approach to online game personalisation provides an effective basis for automatically balancing the game experience in actual video games.

Future work. The present research serves as a demonstrator for online game-space personalisation in actual video games. When implementing the developed personalisation approach in the game INFINITE MARIO BROS. we made the assumption that (1) assessments on exhibited behavioural risk can serve as a proxy for models of user abandonment, (2) the perceived challenge levels monotonically increase with parameter values of the procedural process, and (3) the learned feedback model is constructed from appropriately selected observational features. (Additional) empirical evaluation of these assumptions may further enhance the effectiveness of the approach. In addition, while indicating that human participants consistently prefer the personalised gaming system, additional user studies will decrease experimental variance, and shall further investigate which precise factors lead to a human preferring a personalised system. Indeed, online game personalisation may be considered a multi-objective learning problem in which factors such as challenge, engagement, and frustration need to be balanced.

REFERENCES

[1] D. Charles, M. McNeill, M. McAlister, M. Black, A. Moore, K. Stringer, J. Kucklich, and A. Kerr, "Player-centered design: Player modeling and adaptive digital games," in *DiGRA*, 2005, pp. 285–298.

[2] M. Riedl, "Scalable personalization of interactive experiences through creative automation," *Comp. in Ent.*, vol. 8, no. 4, p. 26, 2010.

[3] R. Petty and J. Cacioppo, "Issue involvement can increase or decrease persuasion by enhancing message-relevant cognitive responses." *Journal of personality and social psychology*, vol. 37, no. 10, p. 1915, 1979.

[4] W. Darley and J. Lim, "The effect of consumers' emotional reactions on behavioral intention: The moderating role of personal relevance and self-monitoring," *Psy. and Mark.*, vol. 9, no. 4, pp. 329–346, 1992.

[5] C. Teng, "Customization, immersion satisfaction, and online gamer loyalty," *Comp. in Hu. Beh.*, vol. 26, no. 6, pp. 1547–1554, 2010.

[6] P. D. Molyneux, "The future of game AI - Lecture," Imperial College London, London, UK, 2006, October 4.

[7] S. C. J. Bakkes, C. T. Tan, and Y. Pisan, "Personalised gaming," *Journal of Creative Technologies*, vol. 3, 2012.

[8] G. N. Yannakakis and J. Togelius, "Experience-driven procedural content generation," *IEEE Tr. Aff. Comp.*, vol. 2, no. 3, pp. 147–161, 2011.

[9] J. K. Olesen, G. N. Yannakakis, and J. Hallam, "Real-time challenge balance in an RTS game using rtNEAT," in *CIG*, 2008, pp. 87–94.

[10] P. H. M. Spronck, I. G. Sprinkhuizen-Kuyper, and E. O. Postma, "Difficulty scaling of game AI," in *GAME-ON*, 2004, pp. 33–37.

[11] R. Hunicke and V. Chapman, "AI for dynamic difficulty adjustment in games," in *AAAI Work. on Ch. in Game AI*, 2004, pp. 91–96.

[12] A. Zook and M. O. Riedl, "A temporal data-driven player model for dynamic difficulty adjustment." in *AIIDE*, 2012.

[13] S. C. J. Bakkes, P. H. M. Spronck, and H. J. Van den Herik, "Rapid and reliable adaptation of video game AI," *IEEE TCIAIG*, vol. 1, no. 2, pp. 93–104, 2009.

[14] J. Togelius, G. N. Yannakakis, K. O. Stanley, and C. Browne, "Search-based procedural content generation: A taxonomy and survey," *IEEE TCIAIG*, vol. 3, no. 3, pp. 172–186, 2011.

[15] J. Fürnkranz, "Recent advances in machine learning and game playing," *ÖGAI-Journal*, vol. 26, no. 2, pp. 19–28, 2007.

[16] H. J. Van den Herik, H. Donkers, and P. H. M. Spronck, "Opponent modelling and commercial games," in *CIG*, 2005, pp. 15–25.

[17] G. Van Lankveld, P. H. M. Spronck, H. J. Van den Herik, and A. Arntz, "Games as personality profiling tools," in *CIG*, 2011, pp. 197–202.

[18] S. C. J. Bakkes, P. H. M. Spronck, and G. van Lankveld, "Player behavioural modelling for video games," *Entertainment Computing*, vol. 3, no. 3, pp. 71–79, 2012.

[19] G. N. Yannakakis and J. Hallam, "Real-time game adaptation for optimizing player satisfaction," *IEEE TCIAIG*, vol. 1, no. 2, pp. 121–133, 2009.

[20] J. Dormans and S. C. J. Bakkes, "Generating missions and spaces for adaptable play experiences," *IEEE TCIAIG*, vol. 3, no. 3, pp. 216–228, 2011.

[21] J. Togelius, M. Preuss, and G. N. Yannakakis, "Towards multiobjective procedural map generation," in *FDG*, 2010, p. #3.

[22] C. Pedersen, J. Togelius, and G. N. Yannakakis, "Modeling Player Experience for Content Creation," *IEEE TCIAIG*, vol. 1, no. 2, pp. 121–133, 2009.

[23] G. Smith, M. Treanor, J. Whitehead, and M. Mateas, "Rhythm-based level generation for 2D platformers," in *FDG*, 2009, pp. 175–182.

[24] J. Togelius, R. De Nardi, and S. M. Lucas, "Making racing fun through player modeling and track evolution," *SAB Workshop on Adaptive Approaches to Optimizing Player Satisfaction*, 2006.

[25] J. Togelius, R. De Nardi, and S. Lucas, "Towards automatic personalised content creation for racing games," in *CIG*, 2007, pp. 252–259.

[26] D.-F. H. Adrian, A. Luisa, and S.-G. Cosio, "An approach to level design using procedural content generation and difficulty curves," in *CIG*, 2013, pp. 1–8.

[27] M. Traichioiu, S. C. J. Bakkes, and D. Roijers, "Difficulty-curve fitting for grammar-based procedural content generation," 2014.

[28] S. C. J. Bakkes and J. Dormans, "Involving player experience in dynamically generated missions and game spaces," in *GAME-ON*, 2010, pp. 72–79.

[29] R. Lopes, T. Tutenel, and R. Bidarra, "Using gameplay semantics to procedurally generate player-matching game worlds," in *PCG*, 2012.

[30] N. Sorenson, P. Pasquier, and S. DiPaola, "A generic approach to challenge modeling for the procedural creation of video game levels," *IEEE TCIAIG*, vol. 3, no. 3, pp. 229–244, 2011.

[31] M. Persson, "Infinite mario," 2009, [Online] Available: http://www.mojang.com/notch/mario.

[32] N. Shaker, G. N. Yannakakis, and J. Togelius, "Crowdsourcing the aesthetics of platform games," *IEEE TCIAIG*, vol. 5, no. 3, pp. 276–290, 2013.

[33] ——, "Digging deeper into platform game level design: session size and sequential features," in *Appl. of Evo. Comp.*, 2012, pp. 275–284.

[34] N. Shaker, S. Asteriadis, G. N. Yannakakis, and K. Karpouzis, "A game-based corpus for analysing the interplay between game context and player experience," in *Aff. Comp. and Int. Inter.*, 2011, pp. 547–556.

[35] C. Pedersen, J. Togelius, and G. N. Yannakakis, "Modeling player experience in super mario bros," in *CIG 2009*, 2009, pp. 132–139.

[36] J. Togelius, S. Karakovskiy, and R. Baumgarten, "The 2009 Mario AI Competition," in *CEC*, 2010, pp. 1–8.

[37] J. Togelius, N. Shaker, and G. N. Yannakakis, "Active player modelling," in *FDG*, 2014.

[38] T. K. Ho, "Random decision forests," in *3rd IEEE Int. Conf. on Document Analysis and Recognition*, vol. 1, 1995, pp. 278–282.

[39] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[40] I. Kononenko, "Estimating attributes: analysis and extensions of relief," in *Machine Learning: ECML-94*, 1994, pp. 171–182.

[41] M. Robnik-Šikonja and I. Kononenko, "Theoretical and empirical analysis of ReliefF and RReliefF," *Machine Learning*, vol. 53, no. 1-2, pp. 23–69, 2003.

[42] G. N. Yannakakis and J. Hallam, "Ranking vs. preference: a comparative study of self-reporting," in *Aff. Comp. and Int. Inter.*, 2011, pp. 437–446.