Circuit-Adaptive Challenge Balancing in Racing Games

Alex Rietveld, Sander Bakkes, and Diederik Roijers
University of Amsterdam
Intelligent Systems Laboratory, Amsterdam, The Netherlands
Email: alex.rietveld@student.uva.nl, s.c.j.bakkes@uva.nl, d.m.roijers@uva.nl

***Abstract*—In this paper, we propose a novel approach to challenge balancing in racing games: *circuit-adaptive challenge balancing*. We propose to automatically adapt the actual racing circuit – while it is being played – such that the performed circuit adaptations intelligently balance the challenge for all players in parallel. Our approach to circuit-adaptive game balancing is submitted as an alternative to the traditional rubber banding method (not a replacement), that may contribute particularly to distinct design goals and gameplay events. Indeed, an interesting feature of the approach, is that each player will be targeted with distinct, player-appropriate circuit adaptations. Consequently, we test the hypothesis that a perceptively balanced game can be achieved via such player-appropriate adaptations. The approach itself is built around (A) a classifier that can assess a player's in-game performance, and (B) an algorithm that employs the ability of targeted circuit adaptations, to the end of realising circuit-based challenge balancing. Experiments that validate the approach – by means of simulation studies and studies with actual human participants – suggest that the approach can automatically balance the challenge in actual racing games, by reducing the gap size between all players in parallel, while ensuring that it does not come at a cost in terms of the actual player experience.**

I. Introduction

In the gaming domain, challenge balancing concerns automatically adapting the challenge that a game poses to the skills of human players [1], [2]. Because a game typically poses a multi-faceted challenge, automatically optimising this challenge to suit individual players is a hard task [3], [4]. A straightforward implementation of challenge balancing exists in many racing games, it is called *rubber banding* [5], [6], [7], [8].

Rubber banding is only informally defined in the academic literature. We use the following definition: rubber banding is artificially reducing the difference in win probability between different players (to a desired level), by changing the capabilities of each character. For instance, the traditional approach to rubber banding ties the speed of the cars to the speed of a particular (human) player [5], i.e., cars in front of the player automatically slow down while those far behind automatically speed up. While such rubber banding may lead to closer races – in which the participants are continuously positioned closely to one another – a well-known disadvantage of the traditional rubber banding technique is that artificially restricting players can feel unfair to the restricted player, and patronizing to the unskilled players [7].

In general, applying too much rubber banding, or using too obvious techniques, can cheat a skilled player out of the crucial feeling of mastery over the game [7]. Therefore, the core idea of the present paper is to be able to balance a racing game such that performed game adaptations are tailored to match the individual player's *performance*. A novel approach for doing so – inspired by research into procedural content generation [9], [10] – is to automatically and continuously *adapt the circuit itself*, to be appropriately balanced to the performance of *all players* of the game in parallel. As such, the goal of our approach to challenge balancing is to (1) ensure a close race, while in parallel (2) tailoring the circuit to the skills of all players (as opposed to restricting player skills, as with the traditional rubber-banding approach). While numerous researchers have studied the online adaptation of racing circuits, to the best of our knowledge, circuit adaptation for the specific task of balancing the provided game challenge has not yet been investigated.

Our approach to circuit-adaptive game balancing is proposed as an alternative to the traditional rubber banding method (not a replacement), that may contribute particularly to distinct design goals and gameplay events. Indeed, an interesting feature of our approach, is that each player will be targeted with distinct, player-appropriate circuit adaptations.

II. Related Work

A. State of the industry

Modern video games typically adopt only simplistic forms of challenge balancing methods. For instance, the online racing mode of the highly popular game GTA V employs a contentious 'catch-up feature' which *noticeably* slows down the leading cars, such that the other players can reach the leading cars more easily. The feature is enabled by default, and has received widespread criticism from game players, who consider the feature "a punishment" [11], [12], [13], "really bad" [14], and "a joke" [15].

The modern racing game PURE expands upon the straightforward rubber banding concept, by creating four groups of opponents directly in front and directly behind the human players, such as to create the *appearance* of the opponents being in direct competition with other opponents in the respective group. It does so by (1) predefining behavioural scripts for the artificial opponents, and (2) dynamically assigning the scripts to groups of opponents [16].

The popular racing game GRID 2 – released in 2013 – implements a circuit-adaptation feature named 'live routes' [17]. It is described as a "new system which gives players unpredictable, dynamically changing routes, ..., which adds a new way to experience racing". The feature, illustrated in

Fig. 1: The racing game GRID 2 randomly adapts the circuit intersections during play of the game.

Figure 1, aims at "provoking the player's driving instincts, driving excitement and producing unexpected gameplay scenarios which keep the action fresh". The live routes feature operates by *randomly* adapting circuit intersections, meaning that the changes do not depend in any way on the players. Furthermore, it is not possible to balance the game for players of distinct skill levels, due to the fact that the circuit changes are applied universally to all players.

B. Academic investigations

Several researchers have investigated the subject of adaptive racing circuits[1]. Togelius *et al*. [18], [19] have explored automatically generating racing circuits which are tailored to match a preference model of a human player. Employing a pre-defined fitness function – reflecting circuit preferences – the goal of the generative system is to evolve a racing circuit that best fits the concerning target model. The work, however, is focussed strictly on circuit generation by itself, not on balancing the challenge in real-time.

Building upon the work of Togelius *et al.* [18], [19], an interactive circuit generator compatible with the open-source racing games TORCS and SPEED DREAMS has been investigated by Cardamone *et al*. [20]. The circuit generator employs a user-guided evolution for automatically generating new, interesting circuits based on the player's evaluation of a population of circuits [21], [22]. When the user is satisfied with the generated circuit, the circuit can subsequently be imported into the actual game. Also this work, however, is focussed strictly on circuit generation by itself, not on balancing the game's difficulty.

Bird *et al*. [23] investigated a fully automatic track generator that works in real time. Based on the performance of the player, their system alternately generates new curves and straights for a track. For instance, if the player is performing well, then sharper curves and shorter, narrower straights will be generated. Experiments indicated that the real-time adapted tracks were better evaluated by players than static tracks, in terms of experienced challenge and fun. A limitation of the developed approach, however, is that its generative process strictly disregards previously generated parts of the track (i.e., it does not create a closed loop); it should be regarded as a track-segment generator, not a circuit generator. As such, the approach is not suitable for implementation in realistic game-play scenarios with multiple players, as game players are generally non-uniformly distributed over the map. Consequently, previously generated parts of the track may not simply be disregarded.

[1]Indeed, the investigation of adaptive racing circuits is often closely related to procedural content generation (PCG). For an extensive overview on (experience-driven) PCG we refer the reader to [10] and [9].

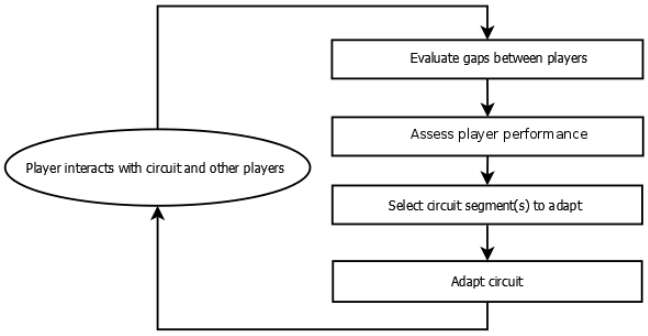


Fig. 2: General procedure of circuit-adaptive challenge balancing. Track adaptations depend on both (1) the classification of the player's performance (as assessed by a SVM classifier), and (2) the relative gaps between the players.

While indeed numerous researchers have studied the online adaptation of racing circuits, to the best of our knowledge, circuit adaptation for the specific task of challenge balancing has not yet been investigated. Our proposed approach to such circuit-adaptive challenge balancing is discussed next.

III. APPROACH

Here, we present our approach to circuit-adaptive challenge balancing for racing games. To provide context, we first describe the racing game that we will employ in our experiments (III-A). Next, we describe a general, minimal framework for circuit adaptation, upon which we base our challenge balancing approach (III-B). Subsequently, we describe two key components of our approach, namely a multi-class linear Support Vector Machine (SVM) [24], [25], [26] classifier that will be trained to classify the player's performance (III-C), and the algorithm that - building on the provided framework for circuit adaptation - performs challenge balancing by considering both (1) the classified player performance, and (2) actual gameplay observations (III-D).

The approach operates on the basis of designer-provided circuit segments, of distinct (intended) challenge level. Offline, an SVM classifier is trained that assesses the player performance per circuit segment. Online, on the basis of the SVM classifications, player-specific circuit-adaptations are performed for all players in parallel. A schematic overview of the approach is given in Figure 2.

A. Game environment

The actual video game in which we will test our approach to circuit-based rubber banding is the open-source racing game DUST RACING 2D [27]. The game, illustrated in Figure 3, can be considered a standard top-down racing game. In the game, the human player can race on distinct circuits, and during play of the game is pitted against numerous (computer-controlled) opponent players. The racing circuits are internally represented as a grid of game objects (track tiles). We have enhanced the game's engine such that (1) it allows the grid to be adapted during play of the game, and that (2) it allows on-demand reloading of the updated grid, so as to create a seamless adaptation of the racing circuit. We will exploit these enhancements for adapting the racing circuit such that distinctly more hard or more easy circuit segments will be

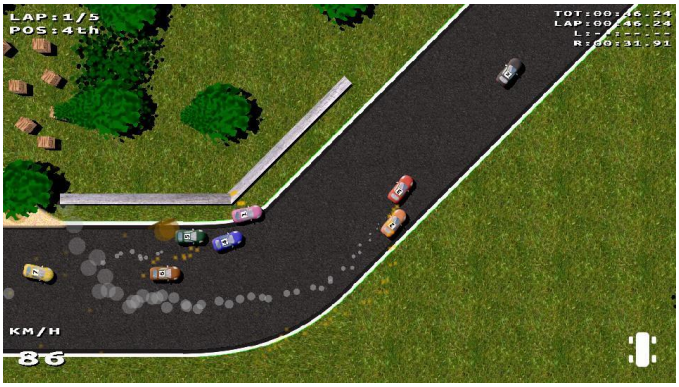


Fig. 3: Screenshot of the game DUST RACING 2D, which is employed in our experiments.

injected as a response to assessments on the observed player performance.

B. Minimal framework for circuit adaptation

Our approach specifically proposes to automatically adapt the actual racing circuit – while it is being played – such that the executed circuit adaptations intelligently balance the challenge for all players of the game in parallel.

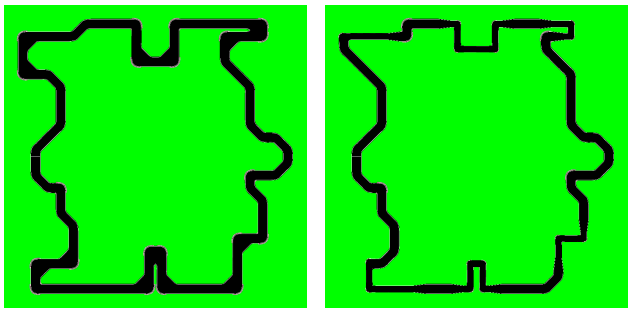
A minimal framework for this approach, that serves as a basis for circuit-based challenge balancing, is as follows. A racing circuit is split up in N segments. For each segment, the game designer provides multiple implementations of the segment; each of predictably distinct challenge. There are numerous track properties which influence how difficult a racing track can be, such as the camber of the road and height variations [28], although for our research we will focus primarily on the width of the track and sharpness of the bends.

During play of the game, the performance of the player(s) is assessed automatically by the game AI.[2] On the basis of this assessment, the game AI intelligently determines (1) which segment(s) should be targeted for adaptation, and (2) which segment implementation should be injected into the targeted circuit segment(s).

A core idea is to adapt circuit segments such that the performed adaptations are tailored to match the individual player's skills. That is, as a guideline, a presently over (under) challenged player will be targeted with circuit segments that predictably match her present performance better. It is important to note that we will achieve this balancing by ensuring that circuit segments are of comparably similar length, while requiring distinct *player skills* to complete them effectively (as opposed to balancing the game by straightforwardly extending select circuit segments such that they require more time to complete).[3] Also, we will ensure that circuit adaptations are seamless, such that they do not interrupt or disturb the game experience for any player. As such, only circuit segments that are currently unoccupied by players will be targeted for adaptation. This requires the number of circuit segments to be sufficiently fine-grained for the specific video game. Figure 4 illustrates the twelve pre-designed level segments which may be injected in the circuit. Figure 5 gives an example of two distinct circuits that are potentially generated by our approach.

[2]While several factors may underlie the observed performance of a player, for the purpose of the present experiments, we consider the observed player performance as an expression of the *skills* of a player.

[3]We here make assumptions on approximations of perceived challenge, while acknowledging that the design of actual, challenging racing circuits is a distinct skill that requires expert (game) designers.



(a) Easiest possible circuit (b) Hardest possible circuit

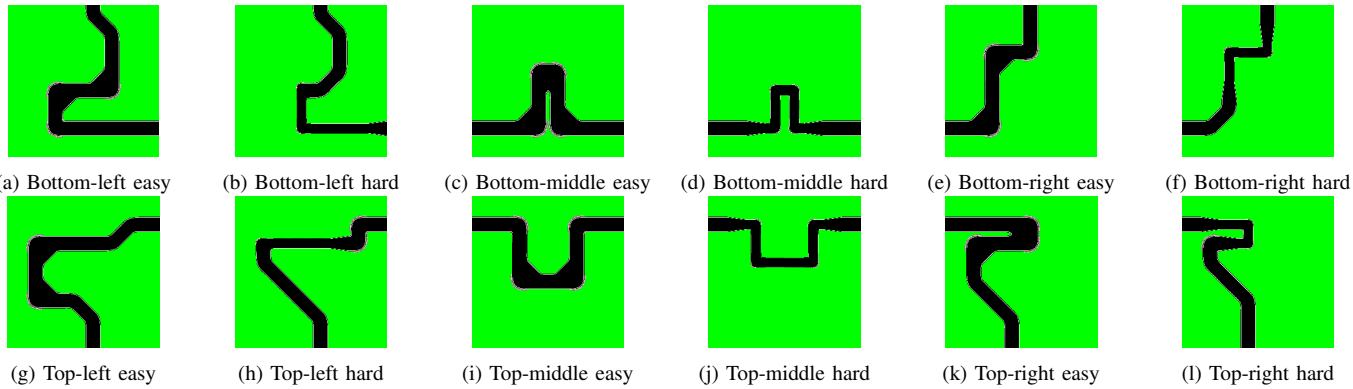
Fig. 5: Example of two generated circuits of distinct challenge level.

Key components within our general framework, are (A) a classifier that can assess a player's performance (it is regarded as an expression of the player's skills), and (B) an algorithm that employs the ability of targeted segment adaptation to the end of realising circuit-based challenge balancing. These components are discussed next.

C. Player-performance classifier

As our goal is to have circuit adaptations intelligently balance the game's *challenge* for all players in parallel, the circuit adaptations need to be performed on the basis of the actual driving performance of the players (as opposed to being performed heuristically using only the player positions, as with traditional rubber banding). As such, we require assessments of the gameplay performance of each individual player. To this end, we will train a classifier that can accurately assess the player's performance in actual gameplay. We train this classifier offline, on the basis of (1) observational data on players interacting with the game, and (2) labels on how well the performances are rated.

Specifically, we will employ a multi-class linear Support Vector Machine (SVM) [24], [25], [26]. In our experience, this classifier has the advantage of being able to work with a limited amount of data; if the soft margin parameter C is appropriately chosen it will have a good out-of-sample generalization. Furthermore, a high classification speed may be expected with this classifier, which is essential for application in actual, time-restrained gaming systems. For training the classifier, we feed the classifier with the provided challenge label, and with data from two high-level observational features. The features, which were selected based on our domain knowledge, are (1) the percentage of time that the player is off track, and (2) the normalised average speed (i.e., how much slower the player is as compared to the obtainable maximum speed). The feature data of the observational gameplay features is normalised to a continuous scale from 0 to 1. The provided performance label is a numeric label $\in [0, 2]$, reflecting a performance that was considered 'under challenging', 'just right', and 'over challenging', respectively.



(a) Bottom-left easy (b) Bottom-left hard (c) Bottom-middle easy (d) Bottom-middle hard (e) Bottom-right easy (f) Bottom-right hard

(g) Top-left easy (h) Top-left hard (i) Top-middle easy (j) Top-middle hard (k) Top-right easy (l) Top-right hard

Fig. 4: The twelve pre-designed level segments which may be injected in the circuit.

To train the SVM classifier we gather labelled data from actual gameplay observations (specifically, we gather labels for each individual circuit segment in actual gameplay). To this end, we present computer-controlled players with a series of short circuit segments, as they will also appear in the actual game. After completing a circuit segment, the computer controlled player labels the challenge on a scale of 'under challenging', 'just right' and 'over challenging'. The provided – simulated – label is stored together with the observed data concerning the two adopted high-level features, being (1) the percentage of time that the player is off track, and (2) the normalised average speed. In the training phase, three distinct computer-controlled players label all circuit segments (i.e., 12 segments in total), over 100 iterations. The provided labels for training the SVM classifier are simulated by the computer-controlled players as follows:

$$\mathtt{label}(\mathtt{obs}[]) = \frac{\sum_{i=1}^{N} \mathtt{obs}[i]}{N} \quad (1)$$

where obs[] denotes a feature vector consisting of gameplay observations (i.e., it contains the two adopted high-level features discussed above), and N reflects the number of observational features (two), of which the values are normalised to a range of 0.0 to 1.0. We heuristically determined that simulated label values below 0.33 translate to the discrete label 'under challenging', values between 0.33 and 0.66 translate to the discrete label 'just right', and values above 0.66 translate to a discrete label 'over challenging'. On the basis of the so-gathered training set, the SVM classifier is built with a heuristically determined soft margin parameter C of 1000.

We note that, given the fact that human-provided labels are a scarce resource, we will always employ a SVM classifier that has been trained with simulated, bot-provided training labels. Acknowledging that this limits the applicability of the classifier for races with humans, we have taken care to program the computer-controlled players to exhibit human-like behaviour such as (probabilistically) missing brake points and steering points. This is described in more detail in Section IV-A.

D. Circuit-adaptive challenge balancing

We will be adapting circuit segments such that the performed adaptations are tailored to match the *individual* player's performance. That is, each player will be targeted with distinct, player-appropriate circuit adaptations. To this end, our circuit-adaptive challenge balancing approach is built upon two input measures, namely (1) the classification of the performance by the players, and (2) the relative gaps between all players. To reduce the gaps occurring between players, our circuit-adaptive method operates as follows (*cf.* Algorithm 1).

When a player has completed a circuit segment, and when the next circuit segment is still unoccupied, we assess if the gap size between a player and the player ahead / the player behind is within a designer specified tolerance. If this is the case, the SVM classifier is employed to steer the segment injections. That is, if the SVM classifier indicates that the observed gameplay reveals a challenge level that is too high (too low) for the player, then a hard (easy) circuit segment is injected, respectively. If the SVM classifier indicates that the observed gameplay reveals an appropriately balanced performance, then the next circuit segment is injected with the same challenge level as that of the current segment. However, in case the gap size is outside of a designer specified tolerance, we heuristically adapt the next circuit segment, to ensure that the gap size is rapidly reduced regardless of the SVM classification. Specifically, when the gap size is outside of the designer specified tolerance, an easy circuit segment is injected in case a player is too far behind, and a hard circuit segment is injected in case a player is too far ahead.

We note that while in our current setup the performed circuit adaptations are clearly noticeable by the players, indeed, game designers may well be able to mask the adaptations to the racing circuits by visual means (e.g., via the placement of dynamic obstacles).

IV. Experiments

We perform two experiments that test our approach to circuit-adaptive challenge balancing in the actual racing game Dust Racing 2D.

A. Experiment 1

In our first experiment, we test how the proposed minimal framework for circuit adaptation performs on the basis of the SVM classifications. It is proof-of-concept study that evaluates the SVM classifier in a one-player setting without any opponent players; i.e. there is only one car on the circuit. The aim of the experiment is to validate circuit adaptations as

ALGORITHM 1: Circuit-adaptive rubber banding.

```
1  while race-not-finished do
2      for all-players do
3          if player-completes-circuit-segment &&
           next-circuit-segment-unoccupied then
4              if gap-size-within-designer-specified-tolerance then
5                  c ← SVMclassifier(gameplay-observations);
6                  switch c do
7                      case too-easy
8                          inject-hard-circuit-segment;
9                      end
10                     case just-right
11                         inject-circuit-segment-with-same-challenge-
                           level-as-current;
12                     end
13                     case too-hard
14                         inject-easy-circuit-segment;
15                     end
16                 endsw
17             else
18                 if player-is-too-far-behind then
19                     inject-easy-circuit-segment;
20                 else
21                     inject-hard-circuit-segment;
22                 end
23             end
24         end
25     end
26 end
```

a means of balancing the challenge for an individual player. The approach is tested against players of distinct skill level. The experiment is performed first as a simulation study, using computer-controlled players, and then repeated as a user study, employing human participants.

1) Experimental setup: As the SVM classifier is a key component in the framework for circuit adaptation, we test it in actual gameplay. That is, to assess the effectiveness of the approach, we will investigate if the circuit adaptations proposed by using solely the SVM classifier recommendations (*cf.* Algorithm 1, line 4 to 15), lead to the circuit being adapted such that the player's performance is more balanced (i.e., the assessed challenge level converges to 'just right' for the player).

In the simulation study, we use the computer-controlled opponents that are included with the game DUST RACING 2D; the opponents are highly effective at playing the game. We have adapted the opponents to exhibit more human-like behaviour with respect to probabilistically making mistakes with regard to (1) missing indicated brake points, and (2) missing steering points. As such, expert computer-controlled players will have a low probability of making such human-like mistakes, while inversely, beginner computer-controlled players have a high probability of making these mistakes. This ensures that, while the bots are technically capable of achieving the same maximum speed as the human player, like the human player their final performance will depend on the mistakes that they will make during the race. As such, we program three bots, a beginner bot, a novice bot, and an expert bot with a missing brake point / steering point probability of 0.16, 0.10, and 0.07 respectively.

The experiment is performed on three circuits, (1) the easiest possible static circuit, (2) the hardest possible static circuit, and (3) the adaptive circuit – which is adapted during gameplay based on assessments of the player performance. To achieve a bias-free starting condition for the adaptive track, in fifty percent of all experimental sessions the adaptive track was initialised with the easiest possible configuration, and fifty percent with the hardest possible configuration. In this experiment, a session ends when the computer-controlled (human) player has completed 100 (5) laps of the circuit, respectively.

TABLE I: Experiment 1 – Player performance of the beginner bot.

Player performance	Static circuit (easy)	Static circuit (hard)	Adapt. circuit
Under challenging	11.1%	0.4%	10.1%
Just right	**64.9%**	28.9%	57.4%
Over challenging	24.0%	70.8%	32.5%

TABLE II: Experiment 1 – Player performance of the novice bot.

Player performance	Static circuit (easy)	Static circuit (hard)	Adapt. circuit
Under challenging	52.8%	10.9%	24.0%
Just right	46.1%	47.1%	**50.6%**
Over challenging	1.1%	42.0%	25.4%

2) Results: The experimental results for the simulation study are given in Table I, II, and III. The listed classification distribution concerns 800 classified instances, i.e., 8 classification moments per lap (each procedural checkpoint) over 100 laps.

Table I reveals that the beginner bot generally considers the static easy circuit 'just right' (64.9% of the cases), the static hard circuit 'over challenging' (70.8% of the cases), and the adaptive circuit 'just right' 57.4% of the cases). The results indicate that the static easy circuit is the most suitable for the beginner bot, although the adaptive circuit effectively balances the performance compared to the static hard circuit.

Table II reveals that the novice bot generally considers the static easy circuit 'under challenging' (52.8% of the cases), the static hard circuit 'just right' (47.1% of the cases), and the adaptive circuit 'just right' (50.6% of the cases). These results indicate that the adaptive circuit is the most suitable of the three for the novice bot. Indeed, even though 'just right' is the mode among the classifications for the static hard track, a very large proportion of the classifications were 'over challenging' (42.0%), creating a slightly unbalanced overall performance.

Table III reveals that the expert bot generally considers the static easy circuit 'under challenging' (82.0% of the cases), the static hard circuit 'just right' (48.3% of the cases), and the adaptive circuit 'just right' (46.3% of the cases). These results indicate that the hardest possible circuit is the most suitable for the expert bot. Indeed, compared to the static easy circuit, the adaptive circuit effectively balances the performance and approximates the ideal balance of the static hard track.

Altogether, the simulation study shows that certain combinations of skill and circuit configurations result in very unbalanced player performances, such as the beginner bot on the hardest circuit and the expert bot on the easiest circuit. The most suitable environments for the beginner and expert bots are the easiest and hardest static circuits respectively. While the adaptive circuit is not capable of matching those balanced performances for these two bots, it does approximate them and significantly improves upon the situations where an inappropriate circuit was selected. Furthermore, for the novice

TABLE III: Experiment 1 – Player performance of the expert bot.

Player performance	Static circuit (easy)	Static circuit (hard)	Adapt. circuit
Under challenging	82.0%	24.8%	31.3%
Just right	17.5%	**48.3%**	46.3%
Over challenging	0.5%	27.0%	22.5%

bot a satisfactory middle ground is found between the easiest and hardest circuits when using the adaptive approach.

There were a total of 11 human participants in the user study. The experimental results are given in Table IV. It shows that both the static easy (55.7%) and static hard (43.9%) circuits were generally considered 'just right' by the participants. The adaptive circuit was also mostly considered as 'just right': 57.6%. However, there is a noticeable difference in the degree of balance, despite 'just right' being the mode in all three different experimental settings. This is most evident with the static hard track, where 43.6% of the classifications were 'over challenging'; almost as much as the 43.9% of 'just right' classifications. The user study confirms the trend observed in the simulation study: the adaptive circuit is able to effectively balance the player's performance.

B. Experiment 2

In our second experiment, we test the proposed circuit-adaptive challenge balancing method in actual game playing conditions with opponent players. The experiment is performed first as a simulation study, using computer-controlled players, and then repeated as a user study, employing human participants.

1) Experimental setup: We test the proposed approach in an actual circuit race with a total of four players operating in the game environment. In case the experiment is performed by a human participant, we employ the same three computer-controlled opponents that were used in Experiment 1. In case the experiment concerns a simulation study, with only bots operating in the game environment, we employ the same three computer-controlled opponents that were used in Experiment 1, plus an additional novice bot with a missing brake point / steering point probability of 0.13. We compare three distinct experimental setups with each other, namely (**A**) Without any challenge balancing, (**B**) Traditional rubber banding, and (**C**) Circuit-adaptive challenge balancing.

The traditional rubber banding method is implemented as follows: every player is assigned a target position of 2, and during the race, if the player is ahead (behind) of the desired target position, the capabilities of the player's car are automatically reduced (increased) by a fixed factor. This ensures that the player in first place is slowed down, whereas the inverse applies to those in last and second-last, allowing the group of players to stay closely together during the race. In our experiments, we employ a speed factor reduction (increase) of 30%, while the acceleration is reduced (increased) by a factor of 20%, similar to the factors used in the classic racing game MARIO KART: DOUBLE DASH [29]. Each race in this experiment consists of 7 laps around the circuit. For the simulation studies, we simulate 20 runs for each experimental trial, and average the results. The human participants only complete one race per experimental trial.

TABLE V: Experiment 2 – Simulation study – Gap size per challenge balancing approach (average).

Approach	Gap first/last	Gap first/second
Without	52.7	15.9
Rubber banding	5.3	0.1
Circuit-adaptive	37.5	12.4

TABLE VI: Experiment 2 – Simulation study – Player performance per challenge balancing approach (average).

Approach	Under challenging	Just right	Over challenging
Without	9.5%	39.8%	50.7%
Rubber banding	9.7%	45.5%	44.8%
Circuit-adaptive	19.3%	49.5%	31.3%

Each of the three approaches are evaluated in terms of (1) its ability to balance the player performances, as indicated by convergence of the player performance to the target challenge level 'just right', and (2) its ability to reduce the gap size between the players. The gap size reflects the number of checkpoints that are employed within the racing circuit (7 per circuit segment, 56 per circuit lap).

2) Results: Table V gives the experimental results for the simulation study on the achieved gap sizes. We observe that the traditional rubber banding approach, as expected, is able to reduce the gap size between the first and last player (first and second player) as compared to when no form of challenge balancing is employed; 5.3 as compared to 52.7 (0.1 as compared to 15.9). Also, the circuit-adaptive approach is able to reduce the gap size between the first and last player (first and second player) as compared to when no challenge balancing is employed; 37.5 as compared to 52.7 (12.4 as compared to 15.9), though this reduction is not as large as that achieved by the traditional rubber banding method. This is an expected result, as circuit-adaptive challenge balancing, contrary to traditional rubber banding, does not imply crude restrictions on the player speeds, but instead adapts the challenge level of specific circuit segments. While this may lead to a smaller gap reduction, we would like to argue that it leads to a much more balanced player challenge, as we will show next.

Table VI gives the experimental results for the simulation study on the achieved player performances. We observe that without any form of challenge balancing, 50.7% of the classifications are 'over challenging' and 39.8% are 'just right'. With traditional rubber banding, 45.5% of the game segments are classified as 'just right', although an almost equal portion of the classifications were 'over challenging': 44.8%. With circuit-adaptive challenge balancing, 49.5% of the game segments are classified as 'just right'. This result indicates that circuit-adaptive challenge balancing substantially improves the balance of the player performance while in the process of successfully reducing the gap size between players (cf. Table V). Traditional rubber banding slightly improves the balance in the simulations, although it remains at an undesirably low level: the performance is still considered 'over challenging' almost as often as it is 'just right'.

Table VII gives the experimental results for the user study on the achieved gap sizes. Here too we observe that circuit-adaptive challenge balancing is able to reduce the gap between

TABLE IV: Experiment 1 – Player performance of the human participants.

	Static circuit (easy)			Static circuit (hard)			Adapt. circuit		
Part. #	Under challenging	Just right	Over challenging	Under challenging	Just right	Over challenging	Under challenging	Just right	Over challenging
1	66.7%	25.0%	8.3%	25.0%	58.3%	16.7%	33.3%	58.3%	8.3%
2	0.0%	25.0%	75.0%	4.2%	0.0%	95.8%	0.0%	25.0%	75.0%
3	29.2%	50.0%	20.8%	12.5%	66.7%	20.8%	12.5%	58.3%	29.2%
4	4.2%	87.5%	8.3%	4.2%	29.2%	66.7%	25.0%	41.7%	33.3%
5	25.0%	75.0%	0.0%	20.8%	54.2%	25.0%	12.5%	75.0%	12.5%
6	45.8%	54.2%	0.0%	4.2%	70.8%	25.0%	29.2%	50.0%	20.8%
7	16.7%	66.7%	16.7%	8.3%	29.2%	62.5%	12.5%	66.7%	20.8%
8	41.7%	58.3%	0.0%	12.5%	58.3%	29.2%	16.7%	66.7%	16.7%
9	37.5%	58.3%	4.2%	29.2%	54.2%	16.7%	29.2%	62.5%	8.3%
10	41.7%	54.2%	4.2%	16.7%	50.0%	33.3%	20.8%	66.7%	12.5%
11	25.0%	58.3%	16.7%	0.0%	12.5%	87.5%	4.2%	62.5%	33.3%
Average	30.3%	**55.7%**	14.0%	12.5%	**43.9%**	43.6%	17.8%	**57.6%**	24.6%

TABLE VII: Experiment 2 – User studies – Gap size per challenge balancing approach (average).

Approach	Final pos.	First/last	First/second	First/human
Without	1.8	58.8	10.9	10.4
Rubber banding	2.2	11.6	0.2	1.6
Circuit-adaptive	1.8	42.1	10.1	8.4

the first and the last player, compared to when no form of challenge balancing is used (42.1 vs. 58.8). Traditional rubber banding on the other hand is once again the approach that reduces the gap the most between the first and the last player (11.6), and also ensures a very tightly contested finish: the gap between first and second place is on average only 0.2 checkpoints. The circuit-adaptive approach also reduces this gap compared to when no challenge balancing is used, albeit marginally: 10.1 vs. 10.9. As expected, traditional rubber banding is the most successful approach at keeping the group of four cars closely together. However, as we will see in the next table, the reduction in gap size comes at a cost in terms of balancing the player challenge.

Table VIII gives the experimental results for the user study on the achieved player performances. We observe that without challenge balancing, on average 42.5% of the player performances are classified as 'just right' and 42.7% as 'over challenging'. With traditional rubber banding, 39.0% of the classifications are 'just right', and the portion of 'over challenging' has risen to 53.7%. With circuit-adaptive challenge balancing, 49.2% of the player performances are classified as 'just right'. These results reveal that circuit-adaptive challenge balancing substantially improves the balance of player challenge compared to the approach without any form of challenge balancing. Furthermore, the traditional rubber banding approach actually worsened the level of challenge for the participants.

V. Discussion

The results of the simulation study reveal that the adaptive track effectively balances the circuit adaptations such, that the resulting player performances approximate the desired challenge level of an individual player. However, it is interesting to observe that the percentage of instances labelled as 'just right' approximates the desired condition, and not substantially improves upon baseline results. This phenomenon can be explained by the behavioural noise of the bots, where a hard circuit segment may occasionally be labelled as being 'over challenging' (while generally the bot would label a hard circuit segment as 'just right'). Consequently, the adaptive system will generate the next circuit segment to be easier, which the expert bot will generally label as being 'under challenging'. This behavioural noise cannot be avoided, and indeed, can be considered analogous to how human game players interact with the gaming system.

It is evident that the size of the adaptive segments directly influences how much balancing can be performed. Employing relatively large segments makes it harder to individually tailor the circuit, as it requires segments to be unoccupied by players. On the other hand, it is hard for a game designer to create relatively short segments of predictably distinct challenge. We believe our implementation, which divides an entire circuit into six adaptive segments, provides a balance in this regard.

Finally, one may consider the proposed circuit-adaptive challenge balancing approach as one that can co-exist with traditional rubber banding (instead of a replacement). Indeed, a game designer may opt to employ traditional rubber banding methods on particular events, and circuit-adaptation methods on other events.

VI. Conclusion and Future Work

In this paper, we proposed a novel approach to challenge balancing in racing games: *circuit-adaptive challenge balancing*. We proposed to automatically adapt the actual racing circuit – while it is being played – such that the performed circuit adaptations intelligently balance the challenge for all players in parallel. Our approach to circuit-adaptive game balancing is submitted as an alternative to the traditional rubber banding method (not a replacement), that may contribute particularly to distinct design goals and gameplay events. Indeed, an interesting feature of the approach, is that each player will be targeted with distinct, player-appropriate circuit adaptations. Consequently, we tested the hypothesis that a perceptively balanced game can be achieved via such player-appropriate adaptations. The approach itself is built around (A) a classifier that can assess a player's in-game performance, and (B) an algorithm that employs the ability of targeted circuit adaptations, to the end of realising circuit-based challenge balancing.

Experiments that validated the approach to circuit-adaptive challenge balancing – by means of simulation studies and

TABLE VIII: Experiment 2 – User studies – Player performance per challenge balancing approach.

	Without challenge balancing			**Traditional rubber banding**			**Circuit-adaptive challenge balancing**		
Part. #	Under challenging	Just right	Over challenging	Under challenging	Just right	Over challenging	Under challenging	Just right	Over challenging
1	30.4%	57.1%	12.5%	5.4%	51.8%	42.9%	28.6%	51.8%	19.6%
2	0.0%	26.8%	73.2%	1.8%	10.7%	87.5%	3.6%	32.1%	64.3%
3	14.3%	55.4%	30.4%	7.1%	44.6%	48.2%	17.9%	55.4%	26.8%
4	7.1%	25.0%	67.9%	1.8%	53.6%	44.6%	12.5%	42.9%	44.6%
5	16.1%	48.2%	35.7%	17.9%	44.6%	37.5%	28.6%	48.2%	23.2%
6	21.4%	42.9%	35.7%	8.9%	46.4%	44.6%	26.8%	50.0%	23.2%
7	3.6%	33.9%	62.5%	1.8%	23.2%	75.0%	10.7%	58.9%	30.4%
8	19.6%	51.8%	28.6%	12.5%	46.4%	41.1%	32.1%	53.6%	14.3%
9	23.2%	60.7%	16.1%	14.3%	53.6%	32.1%	23.2%	57.1%	19.6%
10	19.6%	42.9%	37.5%	8.9%	32.1%	58.9%	28.6%	46.4%	25.0%
11	7.1%	23.2%	69.6%	0.0%	21.4%	78.6%	10.7%	44.6%	44.6%
Average	14.8%	**42.5%**	42.7%	7.3%	**39.0%**	53.7%	20.3%	**49.2%**	30.5%

studies with actual human participants – revealed that (A) the circuit-adaptive approach effectively adapts the racing circuit for an individual player regardless of her skill, so that a suitable level of challenge is provided, (B) the approach effectively adapts the circuit when a group of *differently* skilled players are competing against each other, and (C) the gap sizes between players are reduced via circuit adaptations, as a consequence of provided challenge levels being balanced to individual players in parallel. From these experimental results, we may conclude that our proposed circuit-adaptive approach provides an effective basis for challenge balancing in actual racing games.

For future work, we will investigate (1) how the circuit-adaptive challenge balancing approach can be applied to complex racing circuits, e.g., circuits that are built from procedurally-generated segments of highly dynamic shape (*cf.* Togelius *et al*. [18], [19]), (2) how to even better learn which circuit segments are appropriate to distinct types of game players, and finally (3) how the circuit-adaptive challenge balancing approach is perceived in multi-player settings with only human players.

REFERENCES

[1] 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.

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

[3] S. Bakkes, C. T. Tan, and Y. Pisan, "Personalised gaming," *Journal: Creative Technologies*, vol. 3, 2013.

[4] S. C. J. Bakkes, C. T. Tan, and Y. Pisan, "Personalised gaming: a motivation and overview of literature," in *Proceedings of The 8th Australasian Conference on Interactive Entertainment: Playing the System*. ACM, 2012, pp. 4:1–4:10.

[5] D. Livingstone and D. Charles, "Intelligent interfaces for digital games," in *Proceedings of the AAAI-04 Workshop on Challenges in Game Artificial Intelligence, AAAI Press, Menlo Park, CA., USA*, 2004, pp. 6–10.

[6] S. Miller, "Auto-dynamic difficulty," *Published in Scott Miller's Game Matters Blog (http://dukenukem.typepad.com/game_matters/2004/01/autoadjusting_g.html)*, 2004.

[7] R. J. Pagulayan, K. Keeker, D. Wixon, R. L. Romero, and T. Fuller, "User-centered design in games," *The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications*, pp. 883–906, 2003.

[8] R. Hunicke, "The case for dynamic difficulty adjustment in games," in *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology*. ACM, 2005, pp. 429–433.

[9] 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.

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

[11] Rockstar, "GTA V official forum - Catch-up is so flawed," 2013, https://support.rockstargames.com/hc/communities/public/questions/200671913-Catch-up-is-so-flawed-.

[12] ——, "GTA V official forum - I hate catch-up on GTA games," 2013, https://support.rockstargames.com/hc/communities/public/questions/200846696-I-hate-catch-up-on-gta-races-.

[13] ——, "GTA V official forum - Catch-up must be off by default - please read," 2013, https://support.rockstargames.com/hc/communities/public/questions/200819936-Catch-up-mus-be-OFF-by-default-R-please-read-.

[14] ——, "GTA V official forum - Catch-up feature in racing utter crap," 2013, https://support.rockstargames.com/hc/communities/public/questions/200242018-Catch-up-feature-in-racing-utter-crap.

[15] ——, "GTA V official forum - Catch-up off," 2013, https://support.rockstargames.com/hc/communities/public/questions/200770286-Catchup-off-.

[16] E. Jimenez, "The pure advantage: Advanced racing game ai," 2009. [Online]. Available: http://www.gamasutra.com/view/feature/3920/the_pure_advantage_advanced_.php

[17] Codemasters Community, "Grid 2 Liveroutes." [Online]. Available: http://community.codemasters.com/t5/Codies-Blog/GRID-2-The-LiveRoutes-System/ba-p/160544

[18] 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.

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

[20] L. Cardamone, D. Loiacono, and P. L. Lanzi, "Interactive track generator for torcs and speed dreams." [Online]. Available: http://trackgen.pierlucalanzi.net/

[21] ——, "Interactive evolution for the procedural generation of tracks in a high-end racing game," in *Proceedings of the 13th annual conference on Genetic and evolutionary computation*. ACM, 2011, pp. 395–402.

[22] D. Loiacono, L. Cardamone, and P. L. Lanzi, "Automatic track generation for high-end racing games using evolutionary computation," *Computational Intelligence and AI in Games, IEEE Transactions on*, vol. 3, no. 3, pp. 245–259, 2011.

[23] J. Bird, T. Feltwell, and G. Cielniak, "Real-time adaptive track generation in racing games," in *Proceedings of the 13th International Conference on Intelligent Games and Simulation (GAMEON'2012)*. EUROSIS-ETI, Ghent University, Ghent, Belgium, 2012.

[24] V. N. Vapnik, *The Nature of Statistical Learning Theory*. Springer, 1995.

[25] ——, *Statistical Learning Theory*. Wiley-Interscience, 1998.

[26] ——, "Support vector method for function estimation," Sep. 7 1999, uS Patent 5,950,146.

[27] J. Lind, "Dust racing 2d." [Online]. Available: http://sourceforge.net/projects/dustrac/

[28] L. McMillan, "A rational approach to racing game track design." [Online]. Available: http://www.gamasutra.com/view/feature/134845/a_rational_approach_to_racing_game_.php?print=1

[29] Y. Ohyagi and K. Satou, "Racing game program and video game device," Oct. 9 2007, US Patent 7,278,913. [Online]. Available: http://www.google.com/patents/US7278913