# Circuit-adaptive Rubber Banding for Racing Games

**MSc Thesis**

by
**Alex Rietveld**
May 19th, 2014

under the supervision of
**Sander Bakkes**,
and submitted to the Board of Examiners in partial fulfillment of the requirements for the degree of

**MSc. in Artificial Intelligence**

at the *University of Amsterdam*

Universiteit van Amsterdam

## Abstract

In the gaming domain, challenge balancing concerns automatically adapting the challenge that a game poses to the skills of human players. Automatically optimizing this challenge to an individual player is a hard task. In the genre of racing games, a straightforward implementation of challenge balancing exists, called *rubber banding*. It ties the speed of the cars to the speed of a particular (human) player, so that cars in front automatically slow down while those far behind become faster. This allows the players to stay closely together. However, the straightforward approach to rubber banding is generally considered to negatively affect the game experience; it is often perceived as excessive and unfair.

In this thesis, we propose a novel approach to challenge balancing for racing games: *circuit-adaptive rubber banding*. We propose to automatically adapt the actual racing circuit – while it is being played – such that the performed circuit adaptations intelligently balance the game experience for all players in parallel. We test the hypothesis that a perceptively fair game can be achieved by unobtrusively balancing the game via player-specific circuit adaptations. Our approach is built around (A) a classifier that can assess a player's experienced challenge, and (B) an algorithm that employs the ability of targeted circuit adaptations, to the end of realising circuit-based rubber banding.

Experiments that validate our approach to circuit-adaptive rubber banding – by means of simulation studies and studies with actual human participants – indicate that the approach can balance the game experience in an actual racing game, by reducing the gap size between all players while in parallel improving the player experiences. The traditional rubber banding approach remains the most effective at reducing the gaps between players, although we observe that it does not positively affect the player experiences. In fact, our user study shows that it is often harmful for the player experiences.

We conclude that there is a solid basis for circuit-adaptive rubber banding to be considered as a viable alternative to the traditional approach. However, special care must be taken when designing the racing circuits and it is imperative to have a reliable experience classifier.

**Acknowledgements**

I would like to thank my supervisor Sander Bakkes for his continued support and advice throughout this research. He introduced me to the field of adaptive gaming and helped me decide the topic of this thesis. It has been a pleasure working with him. Similarly, I appreciate the contributions of the teachers who have taught me a great deal during the last few years. I would also like to thank family and friends for supporting me during my academic career. Finally, a special thanks to the group of people who participated in my experiments; I know it may have been quite tedious and tiresome at times. Their efforts will not be forgotten!

# Contents

# Chapter 1

# 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, 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). 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 this thesis is to be able to balance a racing game such that performed game adaptations are tailored to match the individual player's *skills*. 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 skills of *all players* of the game in parallel. As such, the goal of our approach to rubber banding 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 rubber banding the provided game experience has not yet been investigated.

## 1.1 Research questions

Having established that the traditional rubber banding method is often considered unfair, due to its excessive intervening and artificial manipulation of the race results, we must set a number of practical requirements for our proposed circuit-adaptive approach. If fulfilled, these criteria will provide a basis for our claim that circuit-adaptive rubber banding is a viable alternative to the traditional approach. We therefore pose three research questions in this thesis:

1. *Can a racing game effectively adapt its racing track in real time to the skills of a player?*

The first research question concerns the effects of modifying the playing environment of a racing game. A track may not always be suitable for every type of player: if the track is too difficult, then it will be too challenging for lesser skilled players, whereas if the track is too easy, then expert players will not be challenged sufficiently. We want to investigate if the game environment can automatically adapt in real time to the current player, so that the player experience is improved.

2. *Can a racing game effectively adapt its racing track in real time to a group of differently skilled players simultaneously?*

The second research question builds upon the first one by expanding the setting to one involving multiple players driving on the track simultaneously. We want to know if it is possible to balance a racing game for a group of players by intelligently adapting the track, so that both beginner and expert players will be presented with a more suitably challenging environment, thereby improving the experience for all of the players.

3. *Can real time track adaptations be used to reduce the gaps between players?*

The third and last research question concerns the ability of circuit adaptations to reduce the gaps between differently skilled opponents. Expert players are expected to perform better than beginner players, and so widening gaps naturally evolve during a normal race without any interventions. Traditional rubber banding is known to reduce these gaps, and we want to investigate if our proposed approach is also capable of doing so.

We shall answer these research questions via a number of experiments, and the results will then reveal if circuit-adaptive rubber banding is indeed a viable alternative to the traditional approach.

## 1.2 Thesis Outline

The outline of the thesis is as follows: in Chapter 2 related work from both the video game industry and the academic field is discussed. We highlight the similarities with our own research, as well as pointing out their limitations. In Chapter 3, the approach for circuit-adaptive rubber banding is explained in detail. An overview of the framework is provided and a step-by-step description of the circuit-adaptive procedure is specified. Next, numerous experiments and their results are detailed in Chapter 4. After an initial series of experiments, a number of design changes were made and the experiments were repeated. We include both series of experiments in this thesis to illustrate the difference, as well as providing context for discussions later on. Chapter 5 contains the discussion about the results, as well as mentioning various methods on how to build upon and extend the current research. Finally, a conclusion to the research questions and a summary of the thesis is provided in Chapter 6.

# Chapter 2

# Related Work

This chapter highlights numerous related work, some of which has inspired the research of this thesis. First, we discuss a number of games from the industry and then proceed to academic research. Circuit-adaptive rubber banding is a combination of various ideas from the work discussed next.

## 2.1 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].

Similarly, Nintendo's Mario Kart series has regularly used noticeable speed modifiers to balance the game [16]. The opponents are able to abruptly drive faster than the human player, whenever the player is getting too far ahead. Likewise, they will purposely slow down to allow a player to catch up.

The racing game Pure expands upon the straightforward rubber banding concept, by assigning groups of computer opponents a dynamic target distance with respect to the human player. The target distance can be positive or negative, therefore certain opponents will try to stay ahead of the human player while others will attempt to remain behind. The capabilities

Figure 2.1: The racing game GRID 2 randomly adapts the circuit intersections during play of the game.

of the opponents change depending on the difference between the actual and target distances, so that this difference can be minimized. As a consequence, an opponent's capabilities may be improved so that it can better maintain its target distance, even though this opponent may be ahead of the player. Similarly, an opponent behind the human player will be weakened if it is ahead of its target distance. The goal behind this approach is to make the rubber banding less obvious to the human player [17].

The popular racing game GRID 2 – released in 2013 – implements a circuit-adaptation feature named 'live routes' [18]. 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 Figure 2.1, 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.

## 2.2 Academic investigations

Several researchers have investigated the subject of adaptive racing circuits[1]. Togelius *et al.* [19, 20] have explored automatically generating racing circuits

[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].

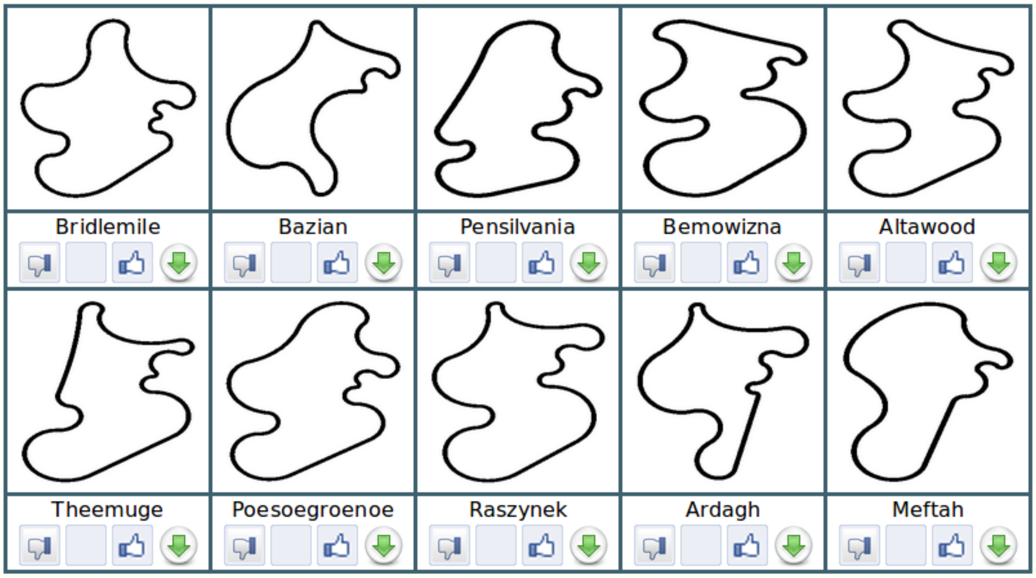



Figure 2.2: The interactive track generator for TORCS and Speed Dreams allows the user to like or dislike the presented tracks. This then influences the newly generated track
.

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 focused strictly on circuit generation by itself, not on rubber banding the player experience in real-time.

Building upon the work of Togelius *et al.* [19, 20], an interactive circuit generator compatible with the open-source racing games TORCS and Speed Dreams has been investigated by Cardamone *et al.* [21]. 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 [22, 23]. When the user is satisfied with the generated circuit, the circuit can subsequently be imported into the actual game. The interface is illustrated in Figure 2.2. Also this work, however, is focused strictly on circuit generation by itself. Furthermore, these generated tracks are not based on any sort of player-model, thus the new track does not necessarily suit the player more appropriately.

Bird *et al.* [24] 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); therefore 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 driving simultaneously, as game players are generally non-uniformly distributed over the map. Consequently, previously generated parts of the track may not simply be disregarded.

While indeed numerous researchers have studied the real time adaptation of racing circuits, to the best of our knowledge, circuit adaptation for the specific task of rubber banding the provided game experience has not yet been investigated. Our proposed approach to such circuit-adaptive rubber banding is discussed in the following chapter.

# Chapter 3

# Approach

The approach to circuit-adaptive rubber banding for racing games is presented next. To provide context, we first describe the racing game that we will employ in our experiments (Section 3.1). Next, we describe a general, minimal framework for circuit adaptation, upon which we base our rubber banding approach (Section 3.2). Subsequently, we describe two key components of our approach to circuit-based rubber banding, namely an SVM classifier that will be trained to classify the actual player experience (Section 3.3), and the algorithm that - building on the provided framework for circuit adaptation - performs rubber banding by considering both (1) the classified player experience, and (2) actual gameplay observations (Section 3.4). A schematic overview of the approach is given in Figure 3.2.

## 3.1 Game environment

The video game which we use to test our approach to circuit-based rubber banding is the open-source racing game Dust Racing 2D [25]. The game, currently still being developed and illustrated in Figure 3.1, can be considered a standard top-down racing game. The car is controlled via the directional arrows on the keyboard, whereby the up-key is used to accelerate, the down-key for braking/reversing and the left and right-keys for steering. The human player can race on various closed-loop 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 track tiles and game objects. The track tiles may consist of a number of different surfaces,


Figure 3.1: Screenshot of the game Dust Racing 2D, which is employed in our experiments.

whereby it is more beneficial for the player, with regards to speed, to keep his car on the road. Checkpoints are placed throughout the circuit to direct the computer-controlled players, as well as preventing the human player from taking shortcuts.

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 rendering of the updated grid, so as to create a seamless adaptation of the racing environment. We will exploit these enhancements to repeatedly adapt the racing circuit such that distinctly more hard or more easy circuit segments will be injected as a response to assessments on the observed player experience.

## 3.2 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 game experience for all players of the game in parallel.

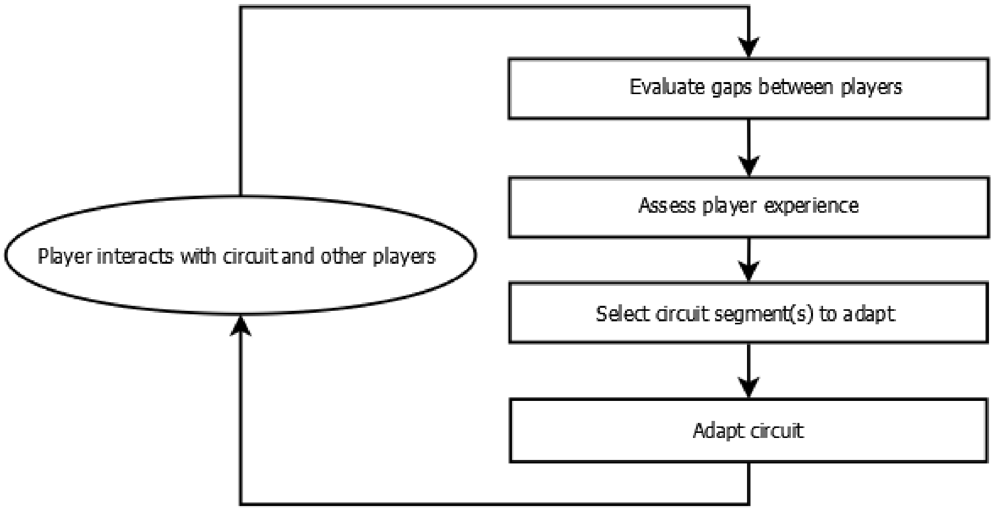
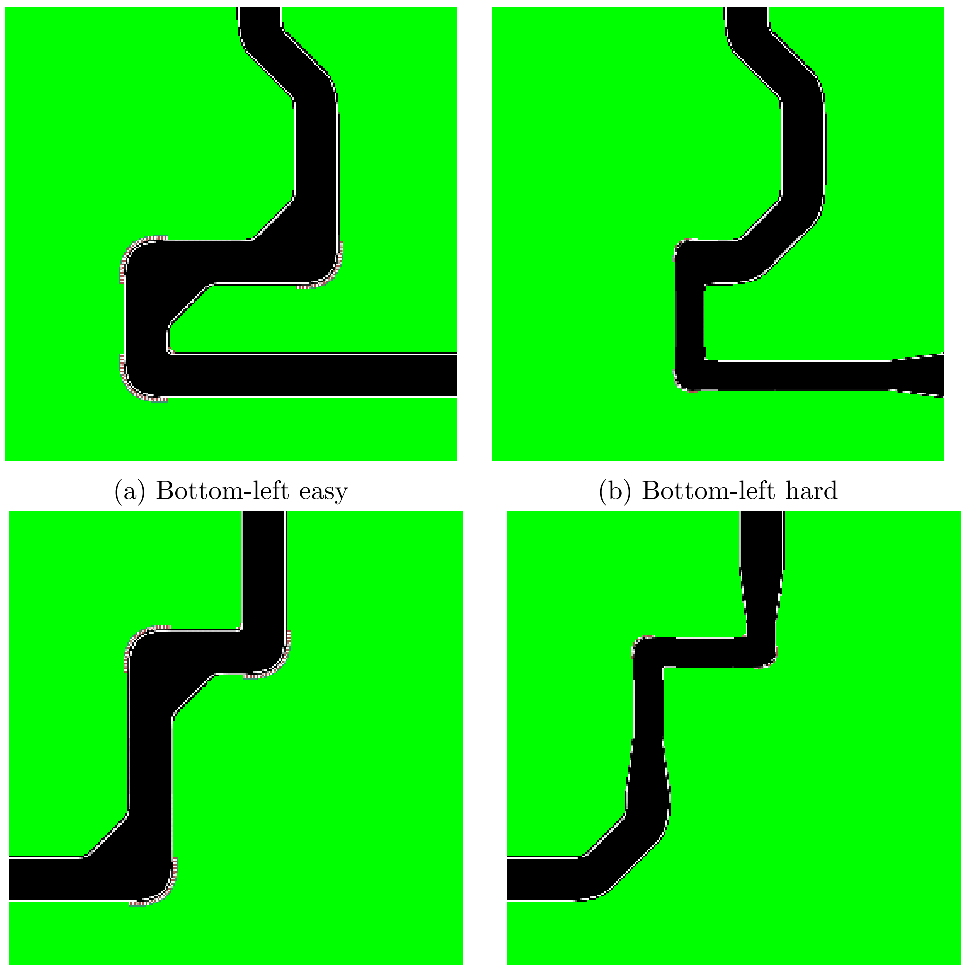



Figure 3.2: General procedure of circuit-adaptive rubber banding. Track adaptations depend on both (1) the classification on the challenge level experienced by the player (as assessed by a SVM classifier), and (2) the relative gaps between the players.

A minimal framework for this approach, that serves as a basis for circuit-based rubber banding, is as follows. A racing circuit is split up in N segments. For each segment, the game designer provides multiple variations of the segment; each of predictably distinct challenge. During play of the game, the challenge level as it is actually experienced by the player(s) is automatically assessed using a classifier, described in further detail in Section 3.3. On the basis of this assessment, the system 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, an expert player will be targeted with circuit segments that predictably match her challenge level, while beginning players will be targeted with less challenging level segments. 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). 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. In our experiments, we use a track consisting of six adaptable segments, each of which has a distinctly easier and harder alternative. A further two static segments were used for optimizing the overall layout of the level. Figure 3.3 illustrates the variants of some of the adaptable segments which may be injected in the circuit.



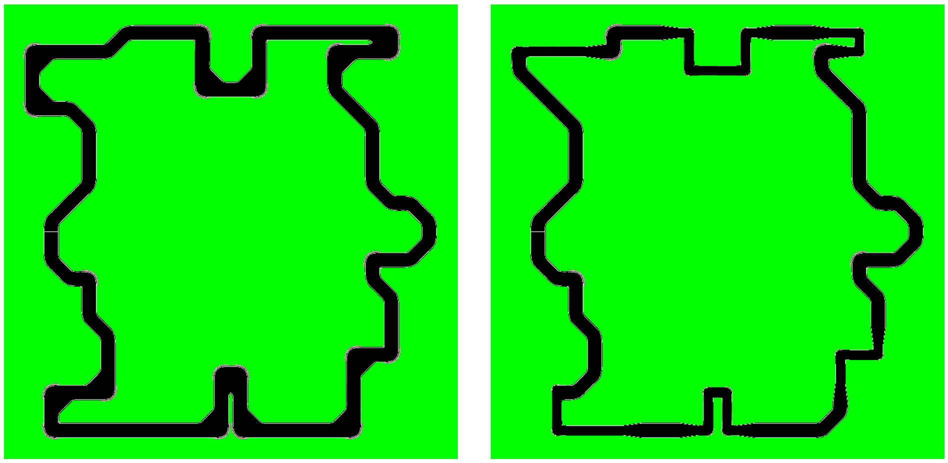
(a) Bottom-left easy

(b) Bottom-left hard

(c) Bottom-right easy

(d) Bottom-right hard

Figure 3.3: An example of the easy and hard variants of some of the level segments which may be injected in the circuit.



(a) Easiest possible circuit

(b) Hardest possible circuit

Figure 3.4: Example of two generated circuits of distinct challenge level. Certain straights and corners of the road are considerably wider in the easier circuit.

It is worth noting that the number of variants per segment does not have to be binary: it is up to the designer to decide how many to use. In Section 5.4 we discuss the number of variants per segment. Figure 3.4 gives an example of two distinct circuits that are potentially generated by our approach, namely the easiest and hardest possible tracks.

Key components within our general framework (illustrated in Figure 3.2), are (A) a classifier that can assess a player's experienced challenge (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 rubber banding. These components are discussed next.

## 3.3 Player-experience classifier

As our goal is to have circuit adaptations intelligently balance the *game experience* for all players in parallel, in our approach, circuit adaptations are performed on the basis of assessments on the actual player experience (as opposed to being solely on the distances between players, as with traditional rubber banding). This requires a player-experience classifier to be trained. In our experiments, we focus on the actual challenge that is experienced by the players. Indeed, the experienced challenge is personal to an individual game player, and may not necessarily coincide with how challenging the developers consider their game. As such, to learn a classifier that can accurately assess the experienced player challenge in actual gameplay, we need to gather (1) observational data on players interacting with the game, and (2) labels on how the players rate their respective experiences.

Specifically, we will employ a multi-class linear Support Vector Machine (SVM) [26, 27, 28]. In our experience, this classifier has the advantage of being able to work with a limited amount of data samples; 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.

To train the SVM classifier we gather labelled data samples, with corresponding data from three 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, (2) the percentage of corners failed (i.e., when a car goes off track near a corner), and (3) the normalised average speed (i.e., how much slower the player is as compared to the obtainable maximum speed). The feature data is normalised to a continuous scale from 0 to 1. The provided challenge label is a numeric label $\in [0, 2]$, reflecting an experience that was considered 'too easy', 'just right', and 'too difficult', respectively.

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 observing a player interact with the circuit segment, we query the player for the experienced challenge. The provided – simulated – label is stored together with the feature data. In the training phase, computer-controlled players label all circuit segments, over 10 iterations. The provided player labels are simulated by the computer-controlled players as follows:

$$\mathsf{label}(\mathsf{obs}[]) = \frac{\sum_{i=1}^{N} \mathsf{obs}[i]}{N}, \tag{3.1}$$

where $\mathsf{obs}[]$ denotes a feature vector consisting of gameplay observations (i.e., it contains the three adopted high-level features discussed above), and $N$ reflects the number of observational features (three), 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 'too easy', values between 0.33 and 0.66 translate to the discrete label 'just right', and values above 0.66 translate to a discrete label 'too difficult'.

Then, we randomly select an equal amount of samples (33 to be precise) from each of the three experienced challenge categories. In other words, the training set for the SVM classifier consists of 33 'too easy' samples, 33 'just right' and finally 33 'too difficult'. The reasoning for using a small subset of the simulated data is discussed in Section 5.5. On the basis of the so-gathered training set, we use cross-validation to build the SVM classifier with a soft margin parameter $\mathsf{C}$ of 1000. The accuracy obtained on the training data was 96.7% and a separate test set of the same size (once again with an equal amount of samples per class) revealed an accuracy of 98.0%. We hereby consider the SVM classifier to be able to accurately predict the experienced challenge based on the feature data.

## 3.4 Circuit-adaptive rubber banding

We will be adapting circuit segments such that the performed adaptations are tailored to match the individual player's skills. To this end, our circuit-adaptive rubber banding is built upon two input measures, namely (1) the classification on the challenge level experienced by the players, and (2) the relative gaps between all players. Figure 3.2 illustrates a schematic overview of the circuit-adaptive rubber banding approach.

A more detailed account of the procedure is as follows: 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 the concerning player is within a designer specified tolerance. Gap sizes are measured by counting the difference in checkpoints obtained so far by each player, and the checkpoints themselves are located along the route, at intervals of approximately equal distance. There are 7 of such checkpoints in each of the 8 segments, meaning that a full lap consists of 56 checkpoints.

If the gap size is within the tolerance, the SVM classifier is employed to steer the segment injections. For each player, the evaluation window for classifying the player-experience consists of the last two completed segments; that is, the SVM classifier exclusively uses game data recorded during the previous two segments. If the SVM classifier indicates that the observed gameplay reveals an experience that is 'too easy' ('too difficult') for the player, then

Algorithm 1. Circuit-adaptive rubber banding.

```
while race-not-finished do
    for all-players do
        if player-completes-circuit-segment && next-circuit-segment-unoccupied
        then
            if gap-size-within-designer-specified-tolerance then
                c ← SVMclassifier(gameplay-observations);
                switch c do
                    case too-easy
                        inject-hard-circuit-segment;
                    end
                    case just-right
                        inject-circuit-segment-with-same-challenge-level-as-
                        current;
                    end
                    case too-hard
                        inject-easy-circuit-segment;
                    end
                endsw
            else
                if player-is-too-far-behind then
                    inject-easy-circuit-segment;
                else
                    inject-hard-circuit-segment;
                end
            end
        end
    end
end
```

a hard (easy) circuit segment is injected, respectively. If the SVM classifier indicates that the observed gameplay reveals an appropriately balanced experience, 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 irregardless 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. The entire circuit-adaptive rubber banding procedure is formally represented in Algorithm 1.

# Chapter 4

# Experiments

We perform two series of experiments that compare circuit-adaptive rubber banding to the traditional approach, as well as comparing both to a baseline without any rubber banding. The first series is comprised of two experiments, namely (1) an experiment that evaluates the effectiveness of circuit adaptation in a single player setting, and (2) an experiment that compares the traditional and circuit-adaptive rubber banding approaches to the baseline, in a multiplayer setting. It evaluates the effects of each approach on the player experiences as well as the size of the gaps between players. The second series repeats the two experiments in an enhanced setting, after observing a number of design issues in the first series.

## 4.1 Series 1: 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 basically evaluates the consequences of circuit adaptation in a one-player setting without any opponent players. This will also serve the purpose of answering the first of our three research questions: *can a racing game effectively adapt its circuit in real time to the skills of a player?* The experiment is performed first as a simulation study, using computer-controlled players, and then repeated as a user study, employing human participants.

### 4.1.1 Experimental setup

As the SVM classifier (configured in Section 3.3 is a key component in the framework for circuit adaptation, we test it in actual gameplay. That is, to assess the performance 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), leads to the circuit being adapted such that the experienced challenge level is more balanced to the player. We consider the experienced challenge to be suitably balanced when the classification 'just right' is the mode among the three different classes. Furthermore, the proportion of 'too easy' and 'too difficult' classifications should simultaneously be minimized. To this end, we use the following formula to compress the distribution of experienced difficulty classifications to a single value, so that it is easier to compare the experimental results:

$$\mathsf{score} = \frac{\mathsf{J} - |\mathsf{E} - \mathsf{D}|}{\mathsf{C}}, \tag{4.1}$$

where $\mathsf{E}$, $\mathsf{J}$ and $\mathsf{D}$ denote the quantity of experience classifications that were 'too easy', 'just right' and 'too difficult' respectively. $\mathsf{C}$ denotes the total quantity of classifications produced by the SVM classifier (i.e. $\mathsf{E} + \mathsf{J} + \mathsf{D}$). In an ideal – albeit utopian – situation whereby the experienced difficulty is *perfectly* balanced for the player, *all* of the classifications would be 'just right'. Using Formula 4.1, this would produce the maximum score of 1. In the worst case, the experienced difficulty is always 'too easy' ('too difficult'), resulting in a score of -1. These two extreme scores will be unlikely though, due to intrinsic randomness within the game as well as the players themselves.

#### Computer controlled players

In the simulation study, we use the computer-controlled players that are included with the game Dust Racing 2D; these AI opponents are highly effective at playing the game. However, we have modified 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 opponent players will have a low probability of making such human-like mistakes, while inversely, beginner opponent 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. The main goal of circuit-adaptive rubber banding is to balance the experienced challenge, and the approach should therefore be applicable to players of varying skill. 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. Each bot will drive around the circuit on its own, analogous to the user study.

#### Racing circuits

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 experience. To achieve a bias-free starting condition for the adaptive circuit, in fifty percent of all experimental sessions the adaptive circuit was initialised with the easiest possible configuration, and fifty percent with the hardest possible configuration. An experimental session ends when the computer-controlled (human) player has completed 100 (3) laps of the circuit, respectively. In series 1, we use our originally designed circuit segments, illustrated in Figure 4.1. As we shall explain later onwards, we redesigned these segments for the enhanced experiments of series 2, so that the difference between easy and hard segments is more noticeable.

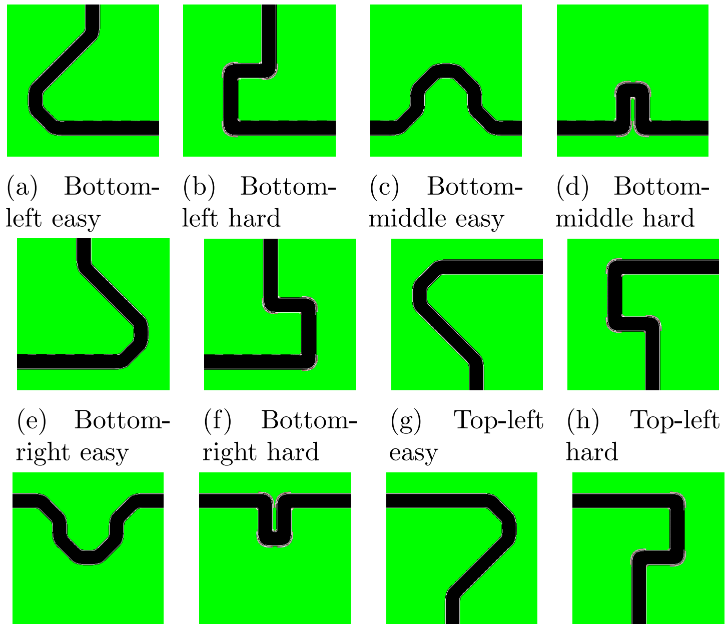
### 4.1.2 Results

The results of experiment 1 are presented next. First, we present the simulation study results and then proceed with our user study.

#### Simulation study

The experimental results for the simulation study are given in Table 4.1, 4.2, and 4.3. The listed classification distribution concerns 800 classified instances, i.e., 8 classification moments per lap (each procedural checkpoint at the end of a segment) over 100 laps.

Table 4.1 reveals that the beginner bot generally considers the static easy circuit ‘just right’ (43.5% of the cases), the static hard circuit ‘too difficult’ (62.3% of the cases), and the adaptive circuit ‘just right’ (48.3% of the cases).



(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

Figure 4.1: The original twelve level segments which may be injected in the circuit.

Using Formula 4.1 to rate the desired level of balance (ranging from -1 to 1, *cf.* Section 4.1.1), the static easy circuit has a score of 0.30, the static hard circuit has a score of -0.25 and finally the adaptive circuit is scored at 0.35. The results indicate that the adaptive circuit effectively balances the circuit, and is the best suited for the beginner bot with respect to the experienced challenge.

Table 4.2 indicates that the novice bot generally considers the static easy circuit 'too easy' (51.0% of the cases), the static hard circuit 'just right' (47.5% of the cases), and the adaptive circuit 'just right' (48.3% of the cases). The scores for the three circuits are -0.02, 0.20 and 0.48 respectively. The results show that the adaptive circuit is (by a large margin) the most suitable of the three for the novice bot.

Table 4.3 shows that the expert bot generally considers the static easy circuit 'too easy' (55.3% of the cases), the static hard circuit 'just right' (53.5% of the cases), and the adaptive circuit 'just right' (53.3% of the cases). The static easy track has a score of -0.11, the static hard track has a score

| Player experience | Static circuit (easy) | Static circuit (hard) | Adaptive circuit |
|---|---|---|---|
| Too easy | 35.3% | 4.1% | 19.1% |
| Just right | 43.5% | 33.6% | **48.3%** |
| Too difficult | 21.3% | 62.3% | 32.6% |

Table 4.1: Experiment 1 – Player experience of the beginner bot.

| Player experience | Static circuit (easy) | Static circuit (hard) | Adaptive circuit |
|---|---|---|---|
| Too easy | 51.0% | 12.4% | 25.5% |
| Just right | 40.4% | 47.5% | **51.4%** |
| Too difficult | 8.6% | 40.1% | 23.1% |

Table 4.2: Experiment 1 – Player experience of the novice bot.

| Player experience | Static circuit (easy) | Static circuit (hard) | Adaptive circuit |
|---|---|---|---|
| Too easy | 55.3% | 16.8% | 24.4% |
| Just right | 37.1% | 53.1% | **53.3%** |
| Too difficult | 7.6% | 29.1% | 19.3% |

Table 4.3: Experiment 1 – Player experience of the expert bot.

of 0.41 and the adaptive circuit scores 0.48. Despite the fact that the static hard track has a marginally higher proportion of 'just right' classifications, the score results indicate that the adaptive circuit creates a better balance of the experienced challenge.

Altogether, the results of the simulation study reveal that regardless of the computer player's aptitude, the adaptive circuit effectively balances the experience according to our scoring system.

### User study

The experimental results for the user study are given in Table 4.4. It reveals that the majority of human participants find the easy circuit 'too easy' (62.5%). Interestingly, the majority of human participants also find the hard circuit 'too easy' (49.1%). In addition, the majority of human participants find the adaptive circuit 'too easy' (49.2%); an analogous result to the player

| | Static circuit (easy) | | | Static circuit (hard) | | | Adaptive circuit | | |
|---|---|---|---|---|---|---|---|---|---|
| Participant # | Too easy | Just right | Too difficult | Too easy | Just right | Too difficult | Too easy | Just right | Too difficult |
| 1 | 62.5% | 37.5% | 0.0% | 70.8% | 20.8% | 8.3% | 66.7% | 33.3% | 0.0% |
| 2 | 87.5% | 12.5% | 0.0% | 75.0% | 25.0% | 0.0% | 75.0% | 20.8% | 4.2% |
| 3 | 12.5% | 50.0% | 37.5% | 20.8% | 58.3% | 20.8% | 8.3% | 50.0% | 41.7% |
| 4 | 87.5% | 12.5% | 0.0% | 45.8% | 50.0% | 4.2% | 58.3% | 33.3% | 8.3% |
| 5 | 62.5% | 25.0% | 12.5% | 33.3% | 41.7% | 25.0% | 37.5% | 37.5% | 25.0% |
| Average | 62.5% | **27.5%** | 10.0% | 49.1% | **39.2%** | 11.7% | 49.2% | **35.0%** | 15.8% |

Table 4.4: Experiment 1 – Player experience of the human participants.

experience on the hard circuit. The average level of balance for the three circuits is -0.25 (static easy), 0.02 (static hard) and 0.02 (adaptive). The adaptive circuit was unable to improve the player experience of the human participants.

The results of the user study imply that the pre-designed level segments that we employ in our approach, pose only limited challenge to the majority of human participants (the circuit segments are not able to substantially increase the challenge level when required). However, circuit segments of additional challenge can always be added to the system. In order to keep the rubber banding experiments simple however, we decided to keep the current two-option circuit segments. Indeed, circuit-adaptive rubber banding on the basis of these segments is still a good idea, as the perceived challenge in a typical human vs. human multi-player setting is generally higher than the perceived challenge in a human vs. computer opponents setting [29]. In Section 4.3 we introduce a number of design changes for the second series of experiments, which address the lack of challenge provided for the human players. For the sake of completeness however, we will first present the results for Experiment 2 using the current format.

## 4.2 Series 1: Experiment 2

In our second experiment of Series 1, we test the proposed circuit-adaptive rubber banding method in actual game playing conditions with opponent players. The experiment is performed first as a simulation study, using exclusively computer-controlled players, and then repeated as a user study, employing human participants to compete against three computer opponents.

The experiment will allow us to answer our second and third research questions (*cf.* Section 1.1).

### 4.2.1 Experimental setup

We test the proposed approach in a regular 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:

A. Without rubber banding.

B. Traditional rubber banding.

C. Circuit-adaptive rubber banding.

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! [30].

As Experiment 1 indicated that the hard circuit is generally more appropriate to the human player (the easy circuit is by a substantial majority found to be 'too easy'), we employ this circuit as the initial circuit configuration when comparing the three approaches. Each race in Experiment 2 consists of 7 laps around the circuit. For the simulation studies, we simulate 20 runs for each experimental setup, and average the results.

The performance of each evaluated approach is expressed in terms of (1) its ability to balance the experienced challenge level, and (2) its ability to

| Approach | Gap first/last | Gap first/second |
|---|---|---|
| Without | 33.1 | 7.4 |
| Traditional | 8.9 | 1.2 |
| Circuit-adaptive | 19.3 | 6.0 |

Table 4.5: Experiment 2 – Simulation study – Gap size per rubber banding approach (average).

reduce the gap size between the players. As mentioned earlier in Section 3.4, the gap size reflects the difference in number of checkpoints obtained at any given moment, and there are 56 per circuit lap.

### 4.2.2 Results

The results for experiment 2 are presented next. Again, we first present the results from our simulation study, followed by the results of the user study.

#### Simulation study

Table 4.5 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 rubber banding method is employed; 8.9 as compared to 33.1 (1.2 as compared to 7.4). Also, circuit-adaptive rubber banding is able to reduce the gap size between the first and last player (first and second player as compared to when no rubber banding method is employed; 19.3 as compared to 33.1 (6.0 as compared to 7.4), though this reduction is not as large as that achieved by the traditional method. This is an expected result, because circuit-adaptive rubber banding, contrary to traditional rubber banding, does not imply crude restrictions on the player speeds, but instead adapts the challenge level. While this may lead to a slightly smaller gap reduction, it leads to a much more balanced experience, as we will show next.

Table 4.6 gives the experimental results for the simulation study on the achieved player experience. We observe that without rubber banding, 45.8% of the game segments during actual gameplay are classified as 'just right'.

| Approach | Too easy | Just right | Too difficult |
|---|---|---|---|
| Without | 16.9% | 45.8% | 37.3% |
| Traditional | 22.7% | 38.6% | 38.7% |
| Circuit-adaptive | 23.9% | 46.8% | 29.3% |

Table 4.6: Experiment 2 – Simulation study – Player experience per rubber banding approach (average).

With traditional rubber banding, 38.6% of the game segments are classified as 'just right'. With circuit-adaptive rubber banding, 46.8% of the game segments are classified as 'just right'. This result indicates that (A) the traditional rubber banding method noticeably and negatively affects the player experience, and (B) circuit-adaptive rubber banding, on the other hand, does not negatively affect the player experience while in the process of successfully reducing the gap size between players (Table 4.5).

**User study**

Table 4.7 gives the experimental results for the user study on the achieved gap sizes. Here too we observe that circuit-adaptive rubber banding is able to reduce the gap between first and last, compared to when no form of rubber banding is used (30.0 vs. 49.0). Traditional rubber banding on the other hand is once again the approach that reduces the gap the most between first and last (24.8).

The results also show that the circuit-adaptive rubber banding approach appears to reduce the gap size between the first and the second player, as compared to the traditional rubber banding approach (a gap size of 9.0 vs. 19.8). However, this result is unexpected, and can be attributed to the following two issues in Series 1 of our experiments: (1) for the traditional rubber banding approach, we initially assigned the target positions for the computer controlled players to be inversely related to their skill (e.g. the best bot had a target position of 4 and the weakest bot had a target position of 1). This meant that the expert bot, who was usually in 2nd place, would constantly drive at a slower pace. We redesigned the traditional rubber banding implementation for Series 2 of our experiments. (2) The user study in

| Approach | Final pos. | Gap first/last | Gap first/second | Gap first/human |
|---|---|---|---|---|
| Without | 2.0 | 49.0 | 29.2 | 8.2 |
| Traditional | 1.6 | 24.8 | 19.8 | 0.0 |
| Circuit-adaptive | 1.2 | 30.0 | 9.0 | 0.8 |

Table 4.7: Experiment 2 – User studies – Gap size per rubber banding approach (average).

experiment 1 showed that the majority of the human participants experienced all of our circuits as too easy.

The combination of these two issues meant that the participants who were fairly adept at the game (experiment 1 suggests this was the case for 3 of the 5 participants) were able to quickly escape the group of computer controlled players and extend the gap size with the pursuers. In the circuit-adaptive rubber banding approach, all the cars have the same acceleration and maximum speed properties, so even though the adept human players were still finishing in first place, the circuit adaptations reduced the gap size.

Table 4.8 gives the experimental results for the user study on the achieved player experience. We observe that without rubber banding, 36.5% of the game segments are classified as 'just right' during actual gameplay. With traditional rubber banding, only 31.5% of the game segments are classified as 'just right'. With circuit-adaptive rubber banding, 37.0% of the game segments are classified as 'just right'. Using Formula 4.1, the approach without rubber banding has an average score of 0.03, the traditional rubber banding has an average score of -0.24 and finally circuit-adaptive rubber banding scores an average of -0.11.

Even though circuit-adaptive rubber banding has approximately the same proportion of 'just right' classifications compared to when no rubber banding at all is used (37.0% and 36.5%), the calculated balance scores indicate that the circuit adaptations slightly worsened the balance. This is because easier circuit segments have occasionally been injected into the track, and as we observed earlier, the hardest static circuit which is used in the approach without any rubber banding, was already considered to be 'too easy' during the individual user study for experiment 1 (49.1% in Table 4.4).

| | Without rubber banding | | | Traditional rubber banding | | | Circuit-adaptive rubber banding | | |
|---|---|---|---|---|---|---|---|---|---|
| Participant # | Too easy | Just right | Too difficult | Too easy | Just right | Too difficult | Too easy | Just right | Too difficult |
| 1 | 77.6% | 22.5% | 0.0% | 75.0% | 25.0% | 0.0% | 70.0% | 30.0% | 0.0% |
| 2 | 75.0% | 25.0% | 0.0% | 87.5% | 12.5% | 0.0% | 85.0% | 15.0% | 0.0% |
| 3 | 7.5% | 45.0% | 47.5% | 20.0% | 60.0% | 20.0% | 20.0% | 60.0% | 20.0% |
| 4 | 55.0% | 42.5% | 2.5% | 87.5% | 12.5% | 0.0% | 52.5% | 40.0% | 10.0% |
| 5 | 27.5% | 47.5% | 25.0% | 40.0% | 47.5% | 12.5% | 50.0% | 40.0% | 10.0% |
| Average | 48.5% | **36.5%** | 15.0% | 62.0% | **31.5%** | 6.5% | 55.5% | **37.0%** | 8.0% |

Table 4.8: Experiment 2 – User studies – Player experience per rubber banding approach.

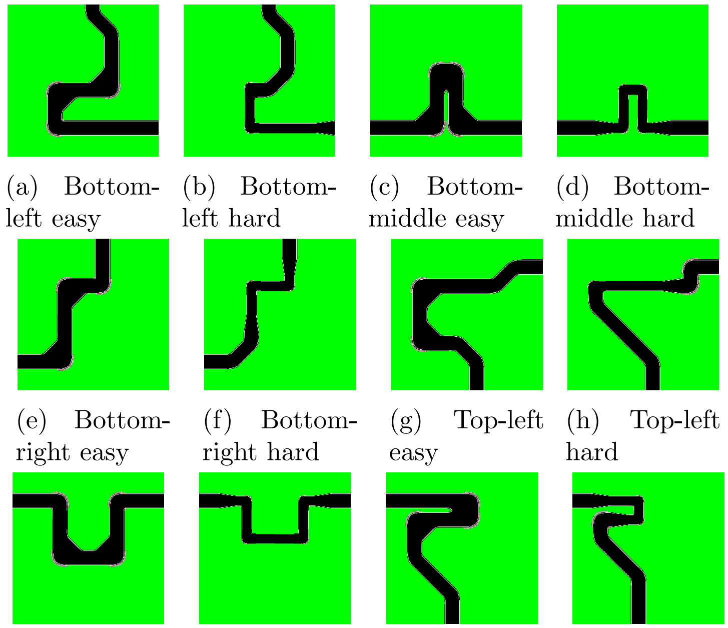
## 4.3 Series 2: Experiment 1

As described in Sections 4.1.2 and 4.2.2, there were a number of issues which hindered our experiments in Series 1. We will first describe these issues and how they were addressed, and then proceed on to the results of the repeated experiments. The first experiment of Series 2 intends to answer our first research question, namely whether circuit adaptations can efficiently be used to balance the game to an individual player.

### 4.3.1 Addressing the issues from Series 1

To begin, it was evident that the majority of the human participants experienced the different circuits – even the hardest possible one – as too easy. As a consequence, the effects of using easier and harder segments was not noticeable enough for the human players when comparing the different approaches to rubber banding. This problem is addressed by first tuning the gameplay so the the overall difficulty of the game is increased: cars now accelerate faster, have less grip on the road and can turn less sharply than before. Secondly, we enhanced the game's track editor so that we had more variety of building blocks to construct our circuits – we added tiles with straights and corners that were wider/narrower than the original tiles packaged in the game.

Then we redesigned all of the circuit segments (the original ones are illustrated in Figure 4.1) so that these too would make the game more difficult, and especially, so that the difference between easier and harder segments was more noticeable. Figure 4.2 illustrates the redesigned circuit segments. Even though the original easier and harder segments appear to be more structurally diverse, the difference was not noticeable enough in our user study



(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

Figure 4.2: The twelve redesigned level segments which may be injected in the circuit.

and one participant even experienced the easier circuit as more challenging than the harder circuit.

One side effect of these changes was that players were significantly more likely to go (marginally) off track around corners. The second high-level observational feature (*cf.* Section 3.3) measures the percentage of corners failed, but even expert players would regularly fail most of the corners at just the slightest contact with a non-road surface. We therefore considered this feature to not be representative for determining a player's experience. Thus, we opted to discard this feature altogether, and retrained the SVM classifier to make the experience classifications based on just the other two features.

Finally, we also tuned the traditional rubber banding implementation so that every car in a multiplayer race would be assigned a target position of 2. Players behind the target position could move faster whereas the car in first place would be slowed down. This ensured a close race at all times and better matched the excessive rubber banding mechanisms used in several video games.

### 4.3.2 Experimental setup

Aside from the aforementioned game design changes, the experiment remains the same. Computer-controlled (human) players complete 100 (3) laps around each circuit and their experience is classified at 8 fixed points in the circuit. That means that we obtain 800 (24) classifications for each (human) player. The same bots from Series 1 are employed.

### 4.3.3 Results

As with the first series of experiments, we conducted a simulation study followed by a user study. The results are presented next.

#### Simulation study

The experimental results for the simulation study are given in Table 4.9, 4.10, and 4.11.

| Player experience | Static circuit (easy) | Static circuit (hard) | Adaptive circuit |
|---|---|---|---|
| Too easy | 11.1% | 0.4% | 10.1% |
| Just right | **64.9%** | 28.9% | 57.4% |
| Too difficult | 24.0% | 70.8% | 32.5% |

Table 4.9: Experiment 1 – Player experience of the beginner bot.

| Player experience | Static circuit (easy) | Static circuit (hard) | Adaptive circuit |
|---|---|---|---|
| Too easy | 52.8% | 10.9% | 24.0% |
| Just right | 46.1% | 47.1% | **50.6%** |
| Too difficult | 1.1% | 42.0% | 25.4% |

Table 4.10: Experiment 1 – Player experience of the novice bot.

Table 4.9 reveals that the beginner bot generally considers the static easy circuit ‘just right’ (64.9% of the cases), the static hard circuit ‘too difficult’

| Player experience | Static circuit (easy) | Static circuit (hard) | Adaptive circuit |
|---|---|---|---|
| Too easy | 82.0% | 24.8% | 31.3% |
| Just right | 17.5% | **48.3%** | 46.3% |
| Too difficult | 0.5% | 27.0% | 22.5% |

Table 4.11: Experiment 1 – Player experience of the expert bot.

(70.8% of the cases), and the adaptive circuit 'just right' 57.4% of the cases). Using Formula 4.1 to rate the desired level of balance (ranging from -1 to 1, *cf.* Section 4.1.1), the static easy circuit has a score of 0.52, the static hard circuit has a score of -0.42 and finally the adaptive circuit is scored at 0.35. The results indicate that the static easy circuit is the most suitable for the beginner bot, although the adaptive circuit effectively balances the experience compared to the static hard circuit.

Table 4.10 indicates that the novice bot generally considers the static easy circuit 'too easy' (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). The scores for the three circuits are -0.06, 0.16 and 0.49 respectively. These results show that the adaptive circuit is (by a large margin) the most suitable of the three for the novice bot. Even though 'just right' is the mode among the classifications for the static hard track, a very large proportion of the classifications were 'too difficult' (42.0%), creating a slightly unbalanced overall experience.

Table 4.11 shows that the expert bot generally considers the static easy circuit 'too easy' (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). The static easy track has a very unbalanced score of -0.64, the static hard track has a score of 0.46 and the adaptive circuit scores 0.41. These results show that the hardest possible circuit is the most suitable for the expert bot. Compared to the static easy circuit, the adaptive circuit effectively balances the experience 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 game experiences, 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

| | Static circuit easy | | | Static circuit hard | | | Adaptive circuit | | |
|---|---|---|---|---|---|---|---|---|---|
| Participant # | Too easy | Just right | Too difficult | Too easy | Just right | Too difficult | Too easy | Just right | Too difficult |
| 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 4.12: Experiment 1 – Player experience of the human participants.

is not capable of matching those balanced experiences for these two bots, it does approximate them and significantly improves upon the situations where an inappropriate circuit was selected. Furthermore, for the novice bot a satisfactory middle ground is found between the easiest and hardest circuits when using the adaptive approach.

### User study

There were a total of 11 human participants for the user studies in Series 2. The experimental results are given in Table 4.12. It shows that both the static easy (55.7%) and 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 'too difficult'; almost as much as the 43.9% of 'just right' classifications. Using Formula 4.1, the balance score for the hard circuit is only 0.13. For the static easy circuit, the score is 0.39 while the adaptive circuit achieves the highest score of 0.51. The user study confirms the trend we observed in the simulation study: the adaptive circuit is able to effectively balance the experience.

## 4.4 Series 2: Experiment 2

Experiment 2 will reveal if circuit-adaptive rubber banding can adapt a circuit to a group of differently skilled players and simultaneously reduce the gaps between these players. This will answer our second and third research questions, described in Section 1.1. We repeat the second experiment from Series 1 using the design changes listed in Section 4.3. Therefore we will once again compare the following three race settings:

A. Without rubber banding.

B. Traditional rubber banding.

C. Circuit-adaptive rubber banding.

### 4.4.1 Results

As before, we first performed a simulation study exclusively with computer-controlled players, and then conducted a user study with the same group of participants from Series 2 Experiment 1.

#### Simulation study

Table 4.13 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 significantly reduce the gap size between the first and last player (first and second player) as compared to when no rubber banding method is employed; 5.25 as compared to 52.7 (0.1 as compared to 15.9). Also, circuit-adaptive rubber banding is able to reduce the gap size between the first and last player (first and second player as compared to when no rubber banding method is employed; 37.5 as compared to 52.7 (12.35 as compared to 15.9), though this reduction is not as large as that achieved by the traditional method. This is an expected result, because circuit-adaptive rubber banding, contrary to traditional rubber banding, does not imply crude restrictions on the player speeds, but instead adapts the challenge level. While this may lead to a smaller gap reduction, it leads to a much more balanced experience, as we will show next.

Table 4.14 gives the experimental results for the simulation study on the achieved player experience. We observe that without rubber banding, 50.7%

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

Table 4.13: Experiment 2 – Simulation study – Gap size per rubber banding approach (average).

| Approach | Too easy | Just right | Too difficult |
|---|---|---|---|
| Without | 9.5% | 39.8% | 50.7% |
| Traditional | 9.7% | 45.5% | 44.8% |
| Circuit-adaptive | 19.3% | 49.5% | 31.3% |

Table 4.14: Experiment 2 – Simulation study – Player experience per rubber banding approach (average).

of the classifications are 'too difficult' 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 'too difficult': 44.8%. With circuit-adaptive rubber banding, 49.5% of the game segments are classified as 'just right'. The balance score for circuit-adaptive rubber banding is significantly better than those obtained by no rubber banding and traditional rubber banding: 0.38 compared to -0.01 and 0.10 respectively.

This result indicates that circuit-adaptive rubber banding significantly improves the player experience while in the process of successfully reducing the gap size between players (Table 4.13). Traditional rubber banding slightly improves the experience in the simulations, although the balance remains at an undesirably low level: the game is still considered 'too difficult' almost as often as it is 'just right'.

### User study

Table 4.15 gives the experimental results for the user study on the achieved gap sizes. Here too we observe that circuit-adaptive rubber banding is able to

| Approach | Final pos. | Gap first/last | Gap first/second | Gap first/human |
|---|---|---|---|---|
| Without | 1.8 | 58.8 | 10.9 | 10.4 |
| Traditional | 2.2 | 11.6 | 0.2 | 1.6 |
| Circuit-adaptive | 1.8 | 42.1 | 10.1 | 8.4 |

Table 4.15: Experiment 2 – User studies – Gap size per rubber banding approach (average).

reduce the gap between first and last, compared to when no form of rubber banding 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 first and last (11.6), and also ensures a very tightly contested finish: the gap between first and second place is on average only 0.2 checkpoints. Circuit-adaptive rubber banding also reduces this gap compared to when no rubber banding is used, albeit very marginally: 10.1 vs. 10.9. Evidently, traditional rubber banding is the most successful at keeping the group of four cars closely together.

Table 4.16 gives the experimental results for the user study on the achieved player experience. We observe that without rubber banding, on average 42.5% of the game segments are classified as 'just right' and 42.7% as 'too difficult'. With traditional rubber banding, 39.0% of the classifications are 'just right', and the portion of 'too difficult' has risen to 53.7%. With circuit-adaptive rubber banding, 49.2% of the game segments are classified as 'just right'. Using Formula 4.1, the approach without rubber banding has on average has a balance score of 0.16, the traditional rubber banding obtains a lower score of -0.07 and finally circuit-adaptive rubber banding scores an average of 0.39. These scores reveal that circuit-adaptive rubber banding on average significantly improved the experience compared to no rubber banding. Furthermore, the traditional rubber banding approach actually worsened the experience for the participants.

| | Without rubber banding | | | Traditional rubber banding | | | Circuit-adaptive rubber banding | | |
|---|---|---|---|---|---|---|---|---|---|
| Participant # | Too easy | Just right | Too difficult | Too easy | Just right | Too difficult | Too easy | Just right | Too difficult |
| 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% |

Table 4.16: Experiment 2 – User studies – Player experience per rubber banding approach.

# Chapter 5

# Discussion

This chapter will discuss a number of topics related to our research and it will end with a comparison of the pros and cons of the different rubber banding approaches.

## 5.1 Portion of 'just right' classifications

The results of the simulation study in Experiment 1 of Series 2 reveal that the adaptive track effectively balances the circuit such, that the resulting gameplay experience approximates the desired challenge level of the player. However, it is interesting to observe that the percentage of instances experienced as 'just right' approximates the desired condition, and does not substantially improve upon the baseline results. This phenomenon can be explained by the behavioural noise of the bots, where a hard (easy) circuit segment may then occasionally be experienced as being 'too difficult' ('too easy') by an expert (beginner) bot, while generally these segments would be experienced as 'just right'. Consequently, the adaptive system will generate the next circuit segment to be easier (harder), which the expert (beginner) bot will then usually consider as being 'too easy' ('too difficult'). This is why in Section 4.1.1 we stated that achieving a maximum balance score of 1 is very unlikely.

The behavioural noise could in theory be minimized, however in that case the bots would not display human like behaviour; instead they would drive along an identical path every single lap. This is undesirable because it would render the simulation study useless, as well as making the bots very predictable for the human participants in Experiment 2.

A possible improvement to our rubber banding system would be to increase the number of experience classes. Currently there are only three, but if there were more (sub)classes then we could predict when a player's experience is threatening to fall beyond the desired 'just right' class, and adapt the circuit as a precaution.

## 5.2 Size of adaptive segments

It is evident that the size of the adaptive segments directly influences how much balancing can be performed, both for the experience and the size of the gaps. Employing relatively large segments makes it harder to individually tailor the circuit, as we require segments to be unoccupied by players before modifying the environment. On the other hand, it is hard for a game designer to create relatively short segments of predictably distinct challenge. Larger segments also make it harder to reduce the gap between players of different skill, as they will both need to drive along the same segment(s) when the cars are close together. Generally, a higher skilled player will tend to perform better than a lesser skilled one on the same circuit segment. We believe our implementation, which uses six adaptive segments, provides a balance in this regard, although a logical next step would be to explore the feasibility of using smaller (micro) circuit segments.

It is also worth pointing out that it is not always necessary for a balancing mechanism to completely reduce the gaps between players. Many (arcade) racing games employ features such as power-ups and drafting. The latter provides a trailing player a small speed boost under the condition that his vehicle is positioned closely enough behind the car in front. Such gameplay elements could be combined with circuit-adaptations to further reduce the gaps.

## 5.3 Multiplayer

In our user studies for experiment 2, we pitted each human participant against three computer-controlled players. Ideally, we would have wanted to let the participants race among themselves, however the game we used does not (yet) support networked multiplayer. There is a split-screen mode but this is restricted to two players and we also believe it would be more practical if every participant had his own screen and keyboard. Nevertheless, a fully functional multiplayer mode without artificial opponents should be explored, to determine if the trends observed in this research are carried over.

## 5.4 Amount of variants per segment

For simplicity, we opted to use only easy and hard alternatives for each adaptive segment. The overall structure of the racing circuit was the same for both, but certain straights and corners had wider roads in the easier variants. As briefly mentioned in Section 3.2 though, the circuit segments do not have to be binary. A designer may choose to create a large quantity of variants per segment, so that circuits can be further personalized to individual players. It is also possible to add segment variants that differ greatly from each other in structure, as long as the overall size of the segment and the connecting points with the previous and following segments remain the same. However, it becomes increasingly difficult to create segments of predictably distinct challenge, even more so when there are significant structural changes.

A solution could be to automatically generate these new segments, thereby relieving the level designer of this burdensome task. When using a multitude of variants per segment, it may be worth considering to procedurally determine which variants are easier and harder for each individual player. What may appear more difficult for one person could in fact seem easier for another. This would remove any potential bias, in the sense that the player's own performance would dictate how the segment variants are ranked among each other. More appropriate segments would subsequently be injected into the circuit when the player's experience is 'too easy' or 'too difficult'.

## 5.5 Alternative classification methods

In our research, we classified a player's experienced using a Support Vector Machine which received as input a small number of observational features (*cf.* Section 3.3). These features were heuristically selected based on our domain knowledge. We initially used several additional metrics, such as the

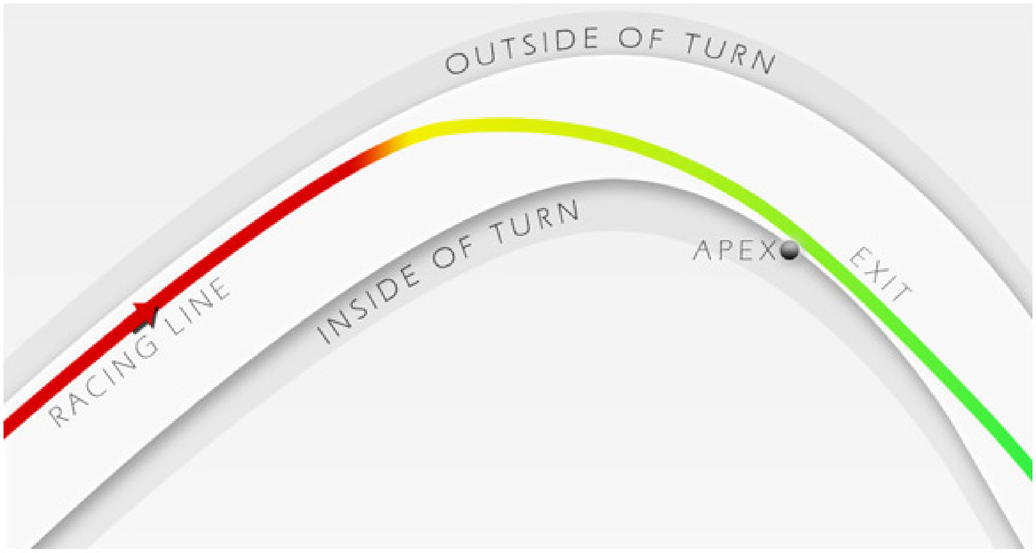



Figure 5.1: The ideal driving line illustrates where a car should position itself on the road, as well as indicating when it should brake (red) and accelerate (green).

*coefficient of variation*, which is defined as the ratio of the standard deviation to the mean. This metric has shown to positively correlate with the frequency of road accidents [31], although we did not observe a significant link with poor driving in our preliminary experiments. We therefore discarded all the non-informative observational features.

Originally, we had planned to obtain our training data for the classifier by asking a small group of human participants in a separate experiment to label their experience upon completing circuit segments. This endeavor was abandoned prematurely though as the participants were far too generous when rating their own experiences. We observed players struggling to remain on the road and battling with the controls, yet still they often labelled their experience as 'just right' or even 'too easy'. The collected samples were clearly unreliable and the resulting SVM classifier was not very useful. We opted to simulate the labels, as a proof of concept that an accurate classifier could be trained using a limited quantity of samples.

A radically different approach of determining how much a player is being challenged is to use a so called *driving line*. In certain racing simulation games, such as the Forza Motorsport series, an in-game system exists which can visually display the optimal path to follow along the racing track.

When activated, it shows where to position the car on the road and when to accelerate and brake. See Figure 5.1 for an example. By measuring the difference between the player's driving line and the optimal driving line (a combination of the difference in position and velocity), it should be possible to classify the experience. When the player's driving line matches (nearly) perfectly with the optimum, it would suggest that the experience is 'too easy': the player has mastered the game. Likewise, if the difference is beyond a given tolerance, then the experience would be deemed 'too difficult'. We believe that this classification approach has the greatest potential, although the main downside is that obtaining the optimal driving line for numerous circuit segments is not at all straightforward.

## 5.6 Comparison of rubber banding approaches

Having completed our experiments with circuit-adaptive rubber banding, we can compose an overview of the method's advantages and disadvantages, in comparison with the straightforward rubber banding method. Table 5.1 sets the main distinguishing points side by side.

As we have seen, main advantages of the traditional rubber banding approach are the fact that it is easier to implement and the consequences are predictable: the gaps between the players will be significantly reduced. Personalized circuits can be obtained with circuit-adaptive rubber banding, although this makes it hard to know *a priori* what the circuit will look like at any given moment of the race. The adaptive circuit can also reduce the size of the gaps, although it is not as efficient as traditional rubber banding. We expect that the use of micro segments could significantly improve the gap reduction of the adaptive circuit, as it would allow the system to make more frequent modifications to the circuit.

The main selling point of circuit-adaptive rubber banding is that it can balance the player experiences for a group of players simultaneously, so that it truly succeeds in challenge balancing. As our experimental results revealed, the traditional rubber banding approach does not improve the player experiences. Furthermore, the approach can be perceived as unfair, as skilled players may frequently be cheated out of a win due to the excessive position interchanging. This makes it very hard to maintain a lead despite driving flawlessly. On the other hand, with circuit-adaptive rubber banding the end result of the race still largely depends on the performance of the players; if a weaker player has caught up to a higher skilled player, it will be because it drove more efficiently on his personalized circuit segments.

The biggest concern for circuit-adaptive rubber banding is that the implementation is not straightforward. Circuit segments must carefully be designed so that they are of predictably distinct challenge and a reliable classification system is required.

| **Traditional** | **Circuit-adaptive** |
|---|---|
| Consequences of rubber banding are predictable for game designer | Circuits can be personalized, but therefore potentially unpredictable circuits |
| Can significantly reduce the gaps and successfully keep the players grouped closely together | Can reduce the gaps, but not as significantly. Using micro segments could help in this regard |
| Easy to understand and implement | Designing suitable circuit segments is not straightforward. A reliable classifier is also required |
| Too rigorous: repeatedly forces position interchanging, thereby punishing flawless performances | End result of race depends on player performances, thereby creating fairer races |
| Does not improve player experiences | Significantly improves player experiences |

Table 5.1: Comparison of the traditional and circuit-adaptive rubber banding approach.

# Chapter 6

# Conclusion

This chapter contains a brief summary of the thesis and provides a conclusion to each of the research questions posed in Section 1.1. The thesis ends with a number of recommendations to extend the research beyond the current investigations.

## 6.1 Summary

Rubber banding is a straightforward implementation of challenge balancing in racing games. It ties the speed of the cars to the speed of a particular (human) player, so that cars in front automatically slow down while those far behind become faster. This allows the players to stay closely together, and the approach is frequently used in the video game industry. However, the straightforward approach to rubber banding is generally considered to negatively affect the game experience; it is often perceived as excessive and unfair, making it too noticeable for the players.

In this thesis, we proposed a novel approach to challenge balancing for racing games: *circuit-adaptive rubber banding*. We proposed to automatically adapt the actual racing circuit – while it is being played – such that the performed circuit adaptations intelligently balance the game experience for all players in parallel. We tested the hypothesis that a perceptively fair game can be achieved by unobtrusively balancing the game via player-specific circuit adaptations. Our approach is built around (A) a classifier that can assess a player's experienced challenge, and (B) an algorithm that employs the ability of targeted circuit adaptations. In our research, player experiences are measured at regular intervals during a race, and they can be classified into one of the following three categories: 'too easy', 'too difficult' and 'just right'. These experiences are used to steer the circuit adaptation; for example, when a player's experience is considered 'too easy' ('too difficult') our system injects a harder (easier) segment into the circuit.

Our classifier consists of a Support Vector Machine that is trained using a small number of observational features abstracted from the game data, while the labels for the training data samples are simulated using computer-controlled players. The system is able to reliably classify the player experiences in real time; this is a crucial property of the classifier.

The primary goal of circuit-adaptive rubber banding is to balance the player's experience, by maximizing the portion of 'just right' classifications, while simultaneously minimizing the other two categories. The secondary goal of circuit-adaptive rubber banding, is to reduce the gaps between players of different skill in a multiplayer setting.

We conducted a number of experiments to compare the different rubber banding approaches. Two series of experiments were presented: our original experiments revealed that there were a number of issues with our implementation, thereby hindering the experimental results. After addressing these design issues, we repeated the experiments. This summary focuses on the results obtained in the second series of experiments.

In experiment 1 we evaluated the circuit adaptations in a single player setting; first by means of a simulation study employing computer-controlled players of different skill, and then with a user study with actual human participants. We compared the adaptive circuit to the easiest and hardest static circuits.

In experiment 2, we compared the following three approaches: (1) no rubber banding, (2) traditional rubber banding and (3) circuit-adaptive rubber banding. The experiment allowed us to compare the experiences of a group of players competing among themselves in a multiplayer setting. It also demonstrated the effects of the different approaches on the size of the gaps between players. Once more, we first conducted a simulation study, consisting exclusively of (differently skilled) computer-controlled players. Then, in the user study we replaced one of the bots with a human participant.

After analyzing the results, a number of alternative implementations were discussed and an overview was presented that compared the main points of circuit-adaptive rubber banding and the traditional rubber banding approach.

## 6.2 Answers to research questions

We will now answer the three research questions formulated in Section 1.1, followed by an overall conclusion to the merit of circuit-adaptive rubber banding.

### 6.2.1 Research question 1

The first research question asked the following:

1. *Can a racing game effectively adapt its racing track in real time to the skills of a player?*

This question concerned the single player setting, whereby a player individually raced along a (adaptive) circuit. Experiment 1 measured the player experiences along the different setups. Both the simulation and user study results revealed that adapting the circuit effectively balances the player experience, compared to when an unsuitable static circuit was used. This is achieved regardless of the player's skill level, thereby confirming our first research question. However, in certain cases the experience of the adaptive circuit only approximated the ideal experience of one of the static, non adaptive circuits. This phenomenon can be attributed to the behavioral noise of both the human and computer-controlled players: occasionally, a player may over-perform (under-perform) on circuit segments that would normally be considered to provide a suitable level of challenge. In Section 5.1 we discussed how a more proactive method of classifying the experience could help maintain the experience at the desired 'just right' level.

### 6.2.2 Research question 2

Our second research question applied to the multiplayer setting, and was formulated as follows:

2. *Can a racing game effectively adapt its racing track in real time to a group of differently skilled players simultaneously?*

Experiment 2 was designed to answer this question, by pitting a group of players against each other and letting them race on an adaptive circuit. A further two setups were also tested (no rubber banding and traditional rubber banding) so that the approaches could be compared. The results indicated that circuit-adaptive rubber banding significantly improved the player experiences compared to the setting without any rubber banding. For the traditional rubber banding approach, the simulation study showed that the player experiences were far from being well-balanced, and in fact the user study revealed that on average the experience was actually worsened compared to when no rubber banding was used. We conclude that circuit-adaptive rubber banding effectively adapted the circuit to a group of differently skilled players.

### 6.2.3 Research question 3

The third and final research question also applied to the multiplayer setting:

3. *Can real time track adaptations be used to reduce the gaps between players?*

This research question was also answered using experiment 2. We compared the three different approaches and measured the average gap sizes among the players. As expected, traditional rubber banding was by far the most effective at reducing the size of the gaps. Circuit-adaptive rubber banding was also capable of reducing the gaps, albeit to a much lower degree. We consider our adaptive approach to be capable of successfully reducing the size of the gaps, however there is much room for improvement. Section 5.2 discussed the option of using smaller (micro) segments, so that the circuit adaptations can be applied more frequently, since we currently require an entire circuit segment to be unoccupied before injecting another variant.

### 6.2.4 General conclusion

From the experimental results, we may conclude that our circuit-adaptive rubber banding approach provides a basis for balancing the game experience in actual racing games, by reducing the gap size between all players while in parallel improving the actual player experiences. As we discussed earlier though, and having experienced it ourselves during our first series of experiments, designing suitable circuit segments of distinct challenge is not straightforward. It is also essential to have a reliable experience classifier. When deciding what type of rubber banding should be used for a specific racing game, one must consider if reducing the gaps is more important than balancing the player experiences.

## 6.3 Future work

An area to explore for future work, is to investigate the possibility of using an alternative classification method for the player experiences. As described in the discussion, we believe *driving lines* (the ideal path to follow along a track) have great potential to determine how difficult the game experience is for a player. However, the main drawback is that this type of feature is not straightforward to implement, and is typically only available in high profile racing simulator games.

We also wish to investigate the idea of (automatically) designing more variants per circuit segment, such as procedurally-generated segments of highly dynamic shape (*cf.* Togelius *et al.* [19, 20]), and how this can be combined with learning how the variants are ranked among each other for each individual player. As such, we would remove the bias from the designer and it would become possible to further personalize the circuit.

Finally, we must investigate if the trends observed in our experiments are also carried over to a fully functional multiplayer setting, whereby each car in a race is controlled by a human player.

# Bibliography


[1] J. K. Olesen, G. N. Yannakakis, and J. Hallam, "Real-time challenge balance in an RTS game using rtNEAT," in *Proceedings of the IEEE 2008 Symposium on Computational Intelligence and Games (CIG'08)*, P. Hingston and L. Barone, Eds., 2008, pp. 87–94.

[2] P. H. M. Spronck, I. G. Sprinkhuizen-Kuyper, and E. O. Postma, "Difficulty scaling of game AI," in *Proceedings of the 5th International Conference on Intelligent Games and Simulation (GAMEON'2004)*, A. E. Rhalibi and D. Van Welden, Eds. EUROSIS-ETI, Ghent University, Ghent, Belgium, 2004, pp. 33–37.

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

[4] ——, "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," *Computational Intelligence and AI in Games, IEEE Transactions on*, vol. 3, no. 3, pp. 172–186, 2011.

[10] G. N. Yannakakis and J. Togelius, "Experience-driven procedural content generation," *Affective Computing, IEEE Transactions on*, 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] Nintendo, "patent for a mario kart game," 2007. [Online]. Available: http://www.google.com/patents/US7278913#v=onepage&q

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

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

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

[20] J. Togelius, R. De Nardi, and S. Lucas, "Towards automatic personalised content creation for racing games," in *Computational Intelligence and Games, 2007. CIG 2007. IEEE Symposium on.* IEEE, 2007, pp. 252–259.

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

[22] ——, "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.

[23] 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.

[24] J. Bird, T. Feltwell, and G. Cielniak, "Real-time adaptive track generation in racing games," in *??? Gameon?*, 2012.

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

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

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

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

[29] D. Clarke and P. R. Duimering, "How computer gamers experience the game situation: A behavioral study," *Comput. Entertain.*, vol. 4, no. 3, Jul. 2006. [Online]. Available: http://doi.acm.org/10.1145/1146816.1146827

[30] 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

[31] M. Taylor, D. Lynam, A. Baruya, and G. Britain, *The effects of drivers' speed on the frequency of road accidents*. Transport Research Laboratory Crowthorne, 2000.