

# Affectively Adaptive Level Generation for Mario

Peter Lehim Pedersen (pleh@itu.dk)  
Christoffer Holmgård Pedersen (holmgard@itu.dk)

Procedural Content Generation in Games, Fall 2012

**Abstract.** In this report, we describe the creation of a first emotionally adaptive level generation system for the Infinite Mario Framework. The system uses continuous physiological readings of sympathetic nervous activity (electrodermal activity and heart rate) as an expression of arousal from interacting with the game. These arousal values are then tied to individual level elements, using an artificial neural network to learn individual players' dispositions toward specific level design elements and use this to procedurally generate personalized levels, predict the player's response and evaluate the actual response to the generated content.

## 1 Introduction

What does it mean for a game to be fun? Enjoyable? Good, even? The subjective experience of emotions related to (video) gameplay are hard to define, quantify and measure objectively.

One possibility is to try and infer the player's engagement with the game, or aspects thereof, and use this as one measure of the quality of the play experience. Engagement can, in turn, with some reduction be operationalized into the player's emotional arousal; a unifying term for the relative sympathetic activation in an individual's nervous system at a particular time.

What in turn stimulates arousal in the player? One obvious answer would be the sum of events that take place over the course of a game, which in turn are unique to each particular game. For classic 2D platform games, the level of the game, in conjunction with the actions NPCs and the player, presents the primary predisposing configuration that determines which events will take place, and hence which level of arousal might be produced in the player.

For this project, we endeavored to create an adaptive level generator for the Infinite Mario Framework. The purpose of the adaptive level generator is to generate individualized levels that predispose the player's experience of a particular 'arousal curve', allowing a designer to predetermine the rise and fall of arousal over the course of a playthrough.

## 2 Background and state of the art

Arousal can be inferred from a multitude of behavioral and physiological signals - indeed the field of affective computing has it as a main goal to establish which

channels and to which degree. Two of the most well-established signals in terms of usefulness for measuring arousal are electrodermal activity and heart rate variability [REFERENCE PICARD HERE].

These physiological signals can be obtained from a game player with relatively little intrusion upon the player and provide responsive measures of arousal that can be gathered in real time [REFERENCE HERE].

Professional game development studios have successfully used electrodermal activity and heart rate variability as measures of arousal and used the signals to provide input to 'AI directors' that manipulate the events of a game to influence player arousal, but only in laboratory settings. A notable example is Valve's work on the Left 4 Dead 2 and Alien Swarm games, where in-game attacks from groups of enemies have successfully been timed to create a roller-coaster like experience of suspense.

Still, it would seem that the technology is not ready for deployment in commercial settings, even though some preliminary moves in that direction have been made by major console makers. In 2011 Sony several filed patents for controllers with embedded physiological measurement components and in 2009 Nintendo announced the development of the Wii Vitality sensor; a peripheral for the Wii console with sensors for electrodermal activity and heart rate variability. Still, none of these products have reached the consumer market to date. This may indicate that the integration of the use of psychophysiological signals into hardware, and perhaps game design, remains a difficult project outside of controlled laboratory conditions and that more research is needed to provide robust and useful methods for using these signals in ways that are relevant for game design. Recent research does, however, indicate that this is an avenue that may be worth pursuing [REFERENCES HECTOR AND GEORGIOS HERE].

Using the data for guiding the procedural selection and generation of content (including, but not limited to event scheduling) for enabling player-adaptive games represents one such opportunity, which we decided to explore for this project.

## 2.1 Game design

The Infinite Mario Framework, is a game derivative of the highly successful 2D sidescrolling platformer series, Super Mario Brothers. The game features a simplistic simulation of gravity that allows Mario to walk, run, and jump - and impressively move midair while in a jump! Using these abilities the object of the game is to guide the protagonist Mario through a level consisting of various obstacles and enemies.

All enemies exhibit simple, consistent behavioral patterns, bringing them close to being moving, deadly obstacles than NPCs proper. As such, the main determinant of the difficulty of any given play session is the configuration of the level.

Though the framework features a relatively small amount of obstacles and enemies to configure each level from, it allows for an practically infinite amount of variation though the placing of these elements and adjustement of the lengths of levels.

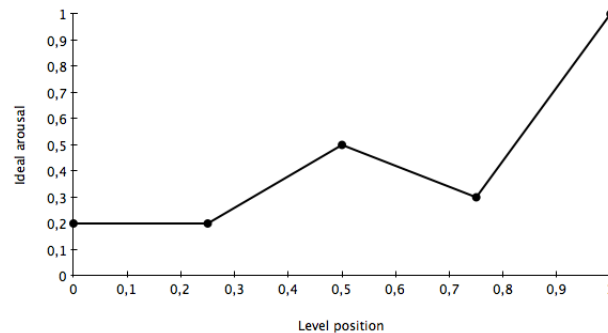
The original Super Mario Brothers games featured levels that were hand crafted by human level designers. This ensures a consistent experience for all players and graded difficulty curve throughout the game, but does not take individual player skill or preference into account.

Since every level is configured by individual components, or tiles, provides a prime opportunity for using procedural content generation to support or replace the human designer.

### 3 Methods

The overall purpose of the bio-level-generator (BLG) that we constructed was to produce levels that were capable of inducing particular patterns of arousal in the player. The ambition was to predict what arousal a particular level feature would induce in a particular player and structure the sequence of features to provide a personal roller-coaster-ride of arousal through a particular level. Ultimately, the solution should allow a designer to specify nothing more than the curve of arousal that she would like the player to experience and the level should be generated adaptively.

For the current project we aimed to make our players have an experience illustrated by the curve in figure ?? . This was our ideal curve for the current version of the level generator and would serve as the later point of reference for the level generator. Thus the challenge was to provide a mapping between level



**Fig. 1.** The intended - or ideal - arousal throughout the level

features and player arousal. To enable this, we produced an experimental setup where each player would play two levels. The first would serve as a benchmark for establishing the player's arousal response to different level features. The second would use this information to generate the adapted level.

### 3.1 Screen-Chunks and Chunks

In order to procedurally generate levels for Mario we needed a method for configuring individual tiles into levels in a way that would motivate gameplay with varying degrees of complexity, with the underlying assumption that this would lead to varying degrees of difficulty and hence engagement and arousal.

Though levels are fundamentally configured of tiles, the individual tile is not necessarily the best level for describing the challenges that the player faces in the game. A tile is quickly traversed by Mario, and the difficulty of doing so can only be determined by looking at neighboring tiles.

Therefore, we decided to understand particular configuration of tiles, or chunks of tiles, as the fundamental unit of challenges that the player faces in the game. The guiding principle for defining a chunk was to look at configurations of tiles that necessitate the player to perform one action, or one combination of actions, to traverse the configuration: for instance jumping over a gap, or onto an enemy to kill it.

Additionally, we defined the concept of screen-chunks: configurations of chunks that are close enough for the actions connected to the chunks to influence each other: E.g a jump over one chunk making Mario land near an enemy that he must kill or avoid.

The fundamental principle of procedurally constructing a level was to connect screen-chunks until a desired level width was achieved. For each screen-chunk, windows for Mario's entry and exit were defined. The windows were required to overlap in neighboring screen-chunks.

The chunks and screen-chunks were designed manually with a custom editor and compiled into a screen-chunk library. It fell to the human designer to ensure that chunks and screen-chunks were passable by the rules of the game. During the design process we tried to ensure a variety of complexity and difficulty in the library, though these evaluations were only based on the expertise of the designer.

Each screen-chunk had a related weight that represented its assumed potential for generating arousal. For random levels this was ignored, while for adapted levels, this was used to select the order of screen-chunks.

**Generating Screen-Chunk Weights** During playthrough of randomly generated levels, arousal responses (as expressed by skin conductivity levels from the player) were continuously sampled at a rate of 30Hz, and mapped to the corresponding screen-chunks.

Before any attempts at adaptivity were done, two reference players were asked to play through a multitude of randomly generated levels, constructing a dataset of 490 observations of arousal responses to screen-chunks distributed across the library. [INDST INFO HER, HVIS VI GIDER] The observations were then used to train an artificial neural network regressor<sup>1</sup> (ANN) to predict arousal values for the individual screen-chunks in the library.

---

<sup>1</sup> A fully connected 40-3-2-1 trained for [HVOR MANGE EPOCHS] epochs

For sessions featuring adaptivity, further, personal, observations were gathered from the first playthrough of a random level. Then, they were used to further train the ANN from it's baseline state toward a personalized state.

Finally, this new adapted ANN was used to predict personal weights for every screen-chunk in the library. This updated library then formed the basis for generating personal, adapted levels.

**SCL sampling and treatment** Udstyr, sample handling The raw samples from of SCL levels were subjected to smoothing by a moving average to remove noise.

### 3.2 Experimental setup

A full play session was comprised of two games of the Mario. The during the first play session, the player was presented with a random level, during the second play session, the player was presented with a personalized level.

In the first playthrough, physiological responses from the player were collected. Subsequently, as detailed in the following section, these measurements were used to construct a weight for each screen-chunk. Each weight represented the arousal that the related screen-chunk would be expected to induce in the player. The weights then informed the construction of the level for the second playthrough.

For random level generation the weights were of course disregarded. For personal level generation they

The concept is illustrated in the Figure

### Signal treatment

**Modeling with an artificial neural network** Neural netvrks-trning -i nye chunk vrdier

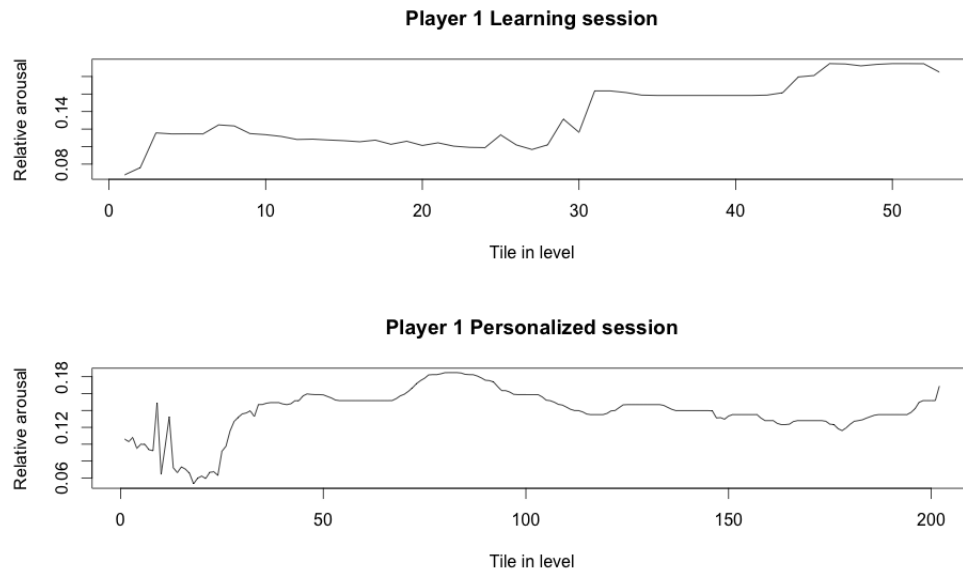
**Intended results** Det sidste i metoden er den 'forventede graf'.

**Classification** ifht. til sb-pcg paper Offline, necessary, parametrized(?), mix of stochastic and deterministic, constructive (but based parameters from person)

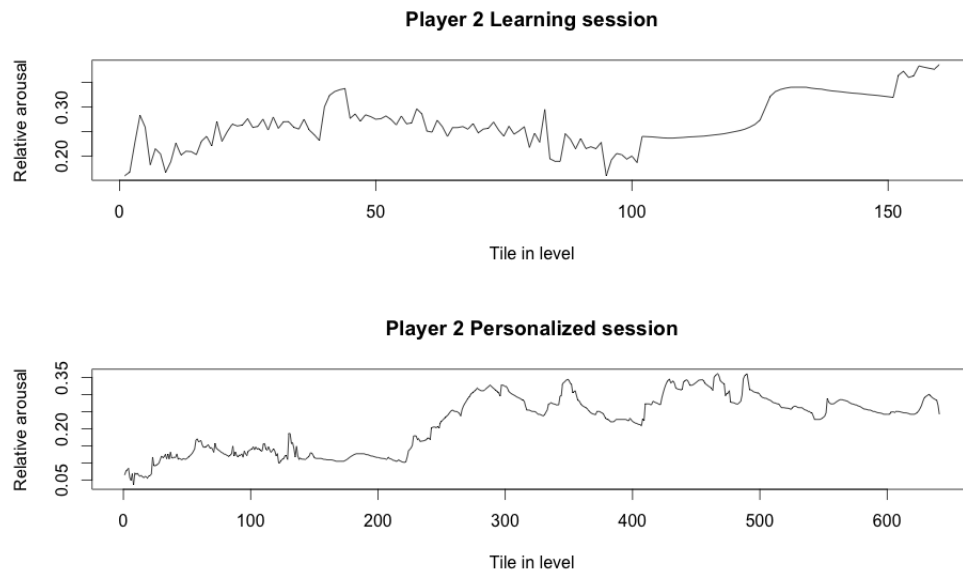
### 3.3 Results

**Building the base model** Statistics about the built model

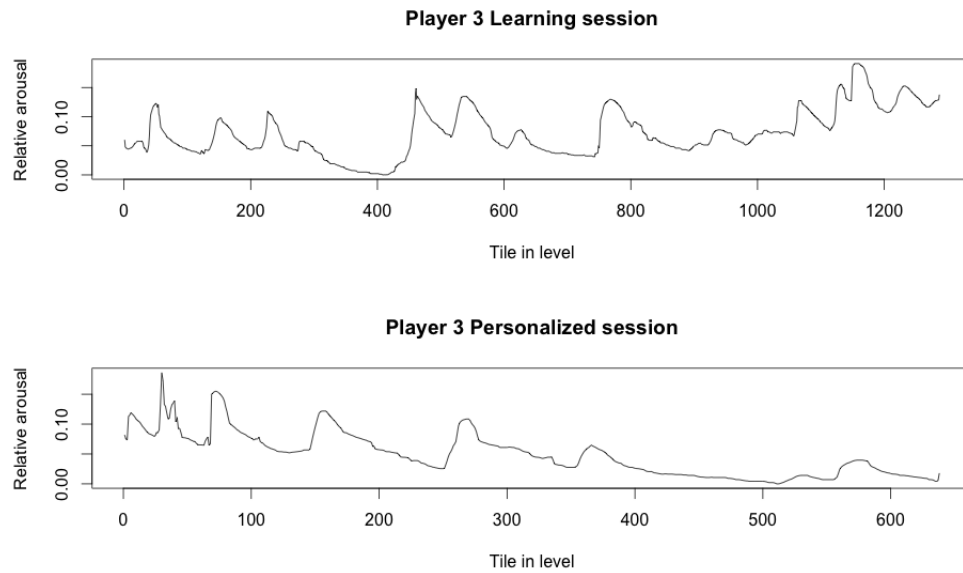
**User testing** Case baseret presentation Learning runs - grafer Personalized runs - grafer



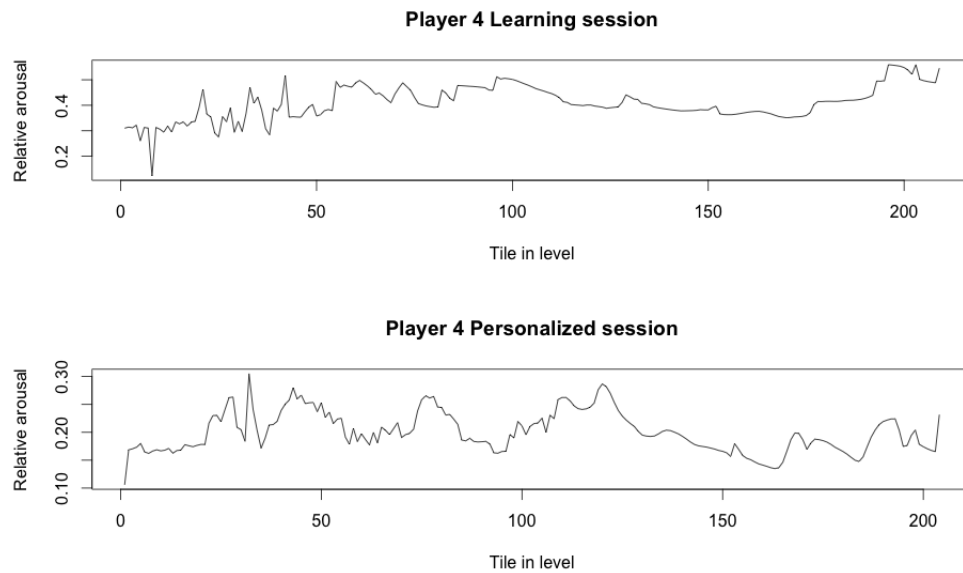
**Fig. 2.** Player 1



**Fig. 3.** Player 2



**Fig. 4.** Player 3



**Fig. 5.** Player 4

## 4 Discussion

Intro

**Arousal-engagement assumptions** Is arousal the same thing as engagement  
- how do we discern between various kinds of arousal; we don't!

**Signal noise** Movement artifacts

**Response lag/screen chunk width** How long time does a response time.  
How long is drop-off of response.

**Habituation/player learning** Skill, learning rate, fatigue

**Player characteristics** Skill, day dependence, relevance of base model,

## 5 Conclusion and future work

Some indications of correlation between expected responses and actual responses.

Some problems in the paradigm.

Insufficient number of observations in the dataset.

More screenshunks would be preferable, but ties in with the size of the dataset.

More advanced signal analysis; that can still run between individual play sessions! (

Need s to be quick; is machine learning an option, really?)

Is this a feasible way of doing PCG? Not commercially yet - will it ever be?

Real-time learning and generation

## References