Chapter 10
The experience-driven perspective (DRAFT)

Noor Shaker, Julian Togelius and Georgios N. Yannakakis

10.1 Nice to get to know you

As you play a game, you get to know it better and better. You understand how to use its core mechanics and how to combine them; you get to know the levels of the game, or, if the levels are procedurally generated, the components of the levels and typical ways in which they can be combined. You learn to predict the behaviour of other creatures, characters and systems in the game. All this you learn from your interaction from the game. While playing, you also adapt to the game: you change your behaviour so as to achieve more success in the game, or so as to entertain yourself better.

However, both you and your game take part in this interaction, and all of your interaction data is available to the game as well. In principle, the game should be able to get to know you as much as you get to know it. After all, it has seen you succeed at overtaking that other car, fail that sequence of long jumps, give up and shut down the game after crashing your plane for the seventh time or finally resort to buying extra moves after almost clearing a particular puzzle. A truly intelligent game should know how you play better than you know it yourself. And then, it should be able to adapt itself so as to entertain you better, or let you achieve more or less success in the game, or perhaps to give you some other kind of experience you would not otherwise have had.

The idea of *game adaptation*, the game adapting itself in response to how you play (or some other information if might have about you), is an old one. In its simplest form it is called “dynamic difficulty adjustment” (DDA), and simply means that the difficulty of the game is increased if the player does well and decreased if the player plays poorly. This can be seen in many car racing games, where the opponent cars always seem to be just ahead of you or just behind you, regardless of how well you play (e.g. rubber banding). The game design rationale for rubber banding is that if the player is much in front of the opponents he/she will not perceive a challenge, and if the player is far behind the opponents he/she will lose hope of ever catching up; in either case, the player will likely lose interest in the game. This is sometimes rationalised as a way of keeping the player in the “flow channel”. *Flow* is a concept which was invented by the psychologist Czikszentmihalyi to signify the “optimal experience”, where someone is completely absorbed in the activity they are performing; one condition for this is constant but not unassailable challenge [**?**]. The flow concept has inspired several theories of challenge and engagement in games, such as GameFlow [**?**] it is, however, rather limited to challenge which is only one of the the multiple concurring dimensions of player experience [3].

DDA mechanisms in racing games are often implemented simply by letting the opponent cars drive faster or slower. There are interesting exceptions, such as the Mario Kart series, which gives more powerful power-ups to players who lag behind, some of which allow them to attack players who lead the pack. Other games might lower the difficulty of a particular section of the game after a player has failed numerous times; *Grand Theft Auto V* allows the player to simply skip any action sequence which the player has failed three times already. There are several proposals for how this could be done more automatically, using AI techniques [**?**]. A key realisation is that adaptation is about more than just difficulty: to begin with, difficulty is multi-dimensional, as a game could be difficult in many different ways, and people have unbalanced skill sets. The same game could be different for player A because of its requirement for quick reactions, for player B because of how difficult the spatial navigation is and for player C because of the nuances of the story that needs to be understood in order to solve its puzzles. Also, just having the right difficulty is in general not enough for a game to be perfectly tailored for a particular player. Different players might prefer different balances of different game elements or atmospheres, such as scary, intense or contemplative parts of the game. Adaptation could in principle happen among any of a multitude of axes, many of which are not properly formalised or even described. There are also many possible methods for adaptation, many of which would include modifying the content of the game or even generating new content.

In this chapter, we will focus on the use of PCG methods to adapt games to the experience of the player. The perspective we adopt is called *experience-driven procedural content generation* [35]. In experience-driven PCG, a model of player experience is learnt, that can predict some aspect of the player’s experience (e.g. challenge, frustration, engagement, spatial involvement, etc.) based on some aspect of game content. This model can then used as a base for an evaluation function in search-based or mixed-initiative PCG. For example, a model might be learned that predicts how engaging some players think individual building puzzles are in a physics-based puzzle game. This model can then be used for evolving new puzzles, where the evaluation function rewards such puzzles that are predicted to be most engaging for the target player(s).

The chapter is structured as follows. First we describe the various ways we can elicit player experience through a game and collect information about player experience. The next section discusses algorithms for creating models of player experience, such as neuroevolutionary preference learning, based on data collected during the game interaction (model's input) and annotated player experience (model's output). A short section discusses how these models can be used in content generation, followed by a prolonged example describing experience-driven level generation in Super Mario Bros in detail.

10.2 Eliciting Player Experience

Games can elicit rich and complex patterns of user experience as they combine unique properties such as rich interactivity and potential for a multifaceted player immersion [3]. User experience in games can be elicited primarily through the long or short term interaction with a list of dissimilar core game elements. Arguably social interaction (or else *shared involvement* [3]) may have a clear impact on a player's experience; however, it offers a rather challenging problem for artificial intelligence and signal processing and experience-driven PCG techniques. While a valid research question for further study social interaction is not included in the list of player experience stimuli considered in this chapter.

Experience-driven PCG views game content as the potential *building block of player experience* [35]. That is precisely the fundamental link between game content and player experience. In that regard, all potential content types can be player experience elicitors. Game content refers to the game environment and its impact to player experience can be directly linked to *spatial involvement* and *affective involvement* as defined by [3] but it also refers to fundamental game design building blocks such as game mechanics (linked to *ludic involvement* [3]), narrative (linked to *narrative involvement* [3]), and reward systems.

Beyond the game environment itself — such as a game level/map [27, 12] — game content includes audiovisual settings such as lighting [5], saturation, and music [6] and sound effects [21] but also virtual camera profiles and effects [**?**, 32, 2] and game rules [28]. The environment usually provides the representation format of stories and narratives and it is also linked to NPCs (if existent in the game) as it forms their context, living habitats and surroundings. In a broader sense, both agents and narratives can be viewed as game content that can be parameterized and altered [35]. Whether games can tell stories [13] or games are instead a form of narrative [**?**] — still an open research question within game studies — stories play an essential part in creating the ambiance, style, climax and feelings of a game. Within the area of interactive storytelling, the story is used as an adaptive mechanism that adjusts according to the actions of the players, offering variant player experiences [**?**]. By breaking the game narrative into further subareas of game content we can find core game content elements such as the games plotline [7], but also the ways this story-plot is represented in the game environment.

In summary, all game content surrounding NPCs (whether those are existent in the game or not) including game mechanics, rules, story-nodes and reward systems may have an effect on the experience of the player [35]. In addition complex, social and emotional non-player characters can be used as triggers of desired player experience. In order for agents to elicit meaningful experience and immerse the player they need to engage players in rich and emotional interaction. Towards that purpose they may embed computational models of cognition, behaviour and emotion which is e.g. based upon the OCC model [**?**].

10.3 Modeling Player Experience

The detection of player experience and the computational modelling of a user's state define core user experience and affective computing problems. Given the complexity and richness of game-player interaction and the multifaceted nature of player experience, methods that manage to overcome the above challenges and model player experience successfully advance our understanding of human behaviour and emotive reaction with human computer interaction. Player Experience Modelling (PEM) can be viewed as a form of user modelling within games incorporating aspects of behaviour, cognition and affect. PEM involves all three key phases for computational model construction including signal processing, feature extraction and selection on the model's input end, experience annotation on the model's output end and variant machine learning and computational intelligence techniques that derive the mapping between the two. Within EDPCG, game content is also represented in the underlying function that characterises player experience.

By clustering the available approaches for PEM we are faced with either *model-based* or *model-free* approaches [35] as well as potential hybrids between them. While a completely model-based approach relies solely on a theoretical framework that maps game context and player responses to experience, a completely model-free approach assumes there is an unknown function between modalities of user input, game content and experience that a machine learner (or a statistical model) may discover which does not assume anything about the structure of this function. The space between a completely model-based and a completely model-free approach can be viewed as a continuum along which any PEM approach might be placed. The rest of this section presents the key elements of both model-based and model-free approaches and discusses the core components of a derived computational model (i.e. model input, model output and common modeling methods).

10.3.1 Model Input and Feature Extraction

The PEM's input can be of three main types: a) player behavioral responses to game content as gathered from **gameplay** data (i.e. behavioural data); b) **objective** data collected as player experience manifestations to game content stimuli such as physiology and body movements; and c) the **game context** which comprises of any type of game content viewed, played through, and/or created [35, 33, 34].

Given the multifaceted nature of player experience the input of a PEM usually consists of complex spatio-temporal patterns which are manifested in the multiple modalities of user input. These signals need to be processed and relevant data features need to be extracted to feed the model. Relevant features, however, are hard to find within such signals and the ad-hoc design of statistical features is often undermining the performance of PEM. There are several available methods within feature extraction (such as principal component analysis and fisher projection) and feature selection (such as sequential forward selection and genetic search based selection) that are applicable to the problem. Most notably, the approaches of *deep learning* [14] and *sequence mining* [16] have shown great potential for the extraction of meaningful features for the task of PEM and for the fusion of data attributes across several player inputs and between player input and game content. In particular, deep learning is a supreme pattern recognition method that manages to detect the most distinct patterns across multiple signals and provides complex spatio-temporal data attributes that complement standard ad-hoc feature extraction [14]. Sequence mining, on the other hand, identifies the most frequent sequences of events across user input modalities and game context which could be relevant as features for any PEM attempt [16].

The three above-mentioned input types for a PEM are detailed in the remaining of this section.

10.3.1.1 Gameplay input

The key motivation behind the use of behavioural (gameplay-based) player input is that player actions and real-time preferences are linked to player experience as games may affect the player's cognitive processing patterns, cognitive focus and emotional state. Arguably it is possible to infer a player's current experience state by analyzing patterns of the interaction and associating player experience with game context variables [4, 8]. The models built on this user input type rely on detailed attributes from the player's behavior which are extracted from player behavioural responses during the interaction with game content stimuli. Such attributes, also named *game metrics*, are statistical spatio-temporal features of game interaction [**?**] which are usually mapped to levels of cognitive states such as attention, challenge and engagement [24]. In general, both generic measures — such as the level of player performance and the time spent on a task — as well as game-specific measures — such as the items picked and used — are relevant for the gameplay-based PEM.

10.3.1.2 Objective input

The variety of available content types within a game can act as elicitators for complex and multifaceted player experience patterns. As expected such patterns of experience may affect changes in the player's physiology, reflect on the player's facial expression, posture and speech, and alter the player's attention and focus level. Monitoring such bodily alterations can assist in recognising and synthesising predictors of player experience. The objective approach to PEM assumes access to multiple modalities of player input which manifest aspects of player experience.Thus, the impact of game content to a number of real-time recordings of the player may be investigated. Physiology offers the primary medium for detecting a player's experience via objective measures: signals obtained from electrocardiography (ECG) [32], photoplethysmography [32, 29], galvanic skin response (GSR) [9], respiration [29], electroencephalography (EEG) [18] and electromyography (among others) are commonly used for the detection of player experience given the recent advancements on sensor technology and physiology-based game interfacing. In addition to physiology the player's bodily expressions may be tracked at different levels of detail and the real-time cognitive or affective responses to game content may be inferred. The core assumption of such input modalities is that particular bodily expressions are linked to basic emotions and cognitive processes. Motion tracking may include body posture [22], facial expression and head pose [24].

Beyond the non-verbal cues discussed above there is also room for verbal cue investigation within games. In general, social signals derived from human verbal communication can potentially be used within social games that allow player-to-player interaction (direct or indirect). Such signals challenge the principles of individual player experience modelling but are expected to open the horizon and augment the potential of the EDPCG framework.

10.3.1.3 Game context input

In addition to gameplay and objective data, the context of the game — e.g. the game content experienced, played or created — defines a necessary input for PEM. Game context refers to the real-time parameterised state of the game which could extend beyond the game content. Without the game context input, player experience models run into the risk of inferring erroneous experience states for the player. For example, an increase in galvanic skin response (GSR) can be linked to a set of dissimilar high-arousal affective states such as frustration and excitement. Thus, the cause of GSR increase (e.g. due to a player's death in a gap between platforms, or alternatively, due to a game level completion) needs to be fused within the GSR signal and embedded in the model. Context-free modelling (while important and desired) has not been investigated to the degree we could identify generic and context-independent content patterns, features and attributes across games and players. A few recent studies, however, such as that of Martinez et al. [15] attempt to investigate context-independent physiological features that can capture player experience across variant game genres.

10.3.2 Model Output: Experience Annotation

The output of the player experience model is provided through an experience annotation process which can either be based on first person reports (self-reports) or on reports expressed indirectly by experts or external observers [35]. The model's output is, therefore, linked to a fundamental research question within player experience and affective computing: what is the ground truth of player experience and how to annotate it? To address this challenge a number of approaches have been proposed; each having benefits and pitfalls. The most direct way to annotate player experience is to ask the players themselves about their experience and build a model based on these annotations. Subjective annotation can be based on either players free-response during play or on forced data retrieved through questionnaires. Alternatively, experts or external observers may annotate the playing experience in a similar fashion. Third-person player experience annotation entails the identification of particular user (cognitive, affective, behavioral) states by user experience and game design experts.

Annotations (either forced self-reports or third-person) can be classified as *rating* (scalar), *class* and *preference* (or ranking). In rating, annotators are asked to answer questionnaire items given in a rating/scaling form — such as the Game Experience Questionnaire [10] or the Geneva Emotion Wheel [1] — which labels user states with a scalar value (or a vector of values). In a class-based format subjects are asked to pick a user state from a particular representation which is usually a simple boolean question (Was that game level frustrating or not? Is this a sad facial expression?). Using the preference annotation format [30], annotators are asked to compare a playing experience in two or more variants/sessions of the game (Was that level more engaging that this level? Which facial expression looks happier?). Recent comparative studies have exposed the limitations of rating approaches over ranking questionnaire schemes which include increased order of play and inconsistency effects [31] and lower inter-rater agreement [17].

10.3.3 Modelling approaches

The approach for constructing models of player experience heavily relies on the modelling approach followed (model-based vs. model-free) and the annotation scheme adopted. For the model-based approach components of the model and any parameters that describe them are constructed in an ad-hoc manner and, sometimes, tested for validity in a trial and error basis. No machine learning or sophisticated computational tools are required for these approaches even though one could envisage the optimisation of the parameter space to yield more accurate models; that, however, would require empirical studies which brings the approach closer to a model-free perspective.

Model-free approaches, on the other hand, are dependent on the annotation scheme and, in turn, the type of model output available. If data recorded includes either a scalar representation (e.g. via ratings) or classes of annotated labels of user states any of a large number of machine learning (regression and classification) algorithms can be used to build affective models. Available methods include artificial neural networks, Bayesian networks, decision trees, support vector machines and standard linear regression. Alternatively, if experience is annotated in a ranked format standard supervised learning techniques are inapplicable, as the problem becomes one of preference learning [30]. Neuro-evolutionary preference learning [30] and rank-based support vector machines [11] but also simpler methods such as linear discriminant analysis [29] are some of the available approaches for learning preferences.

10.4 Content generation through player experience models

The ultimate goal of constructing models of player experience is to use these models as measures of content quality and consequently, realize affective, cognitive and behavioral interaction in games and generate personalized or player-adapted content. Quantitative models of player experience can be used to capture player-game interaction and the impact of game content on player experience. According to Yannakakis and Togelius [**?**] models of player experience can be used to assess content quality and achieve game adaptation.

10.5 Example: Super Mario Bros

A successful complete implementation of the EDPCG framework suggested by Yannakakis and Togelius [**?**] is the work done by Shaker et al. [26, 24, 25] to model and personalize player experience in Infinite Mario Bros (IMB) [20] — a public domain clone of Super Mario Bros [19]. In this work, models of player experience are built based on information collected from the interaction between the player and the game. Different types of features capturing different aspects of player behavior are considered: *subjective* self-reports of player experience; *objective* measures of player experience are collected by video recording gameplay sessions and later extracting information about head movement behavior in reaction to game events [23, 24] (Figures 10.1, 10.2 and 10.3 present example instances of players' reaction when losing, winning and when encountering hard situations, respectively); players' actions while playing the game are also registered and used as *gameplay* features [26].

The choice of feature representation is vitally important since it allows capturing different dimensions of player experience and game content. Furthermore, the choice of content representation defines the search space that can be explored and it affects the efficiency of the content creation method. To accommodate for this, the different sets of features collected are represented as frequencies describing the

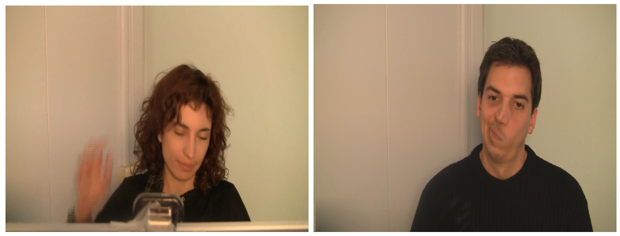

Fig. 10.1: Typical player responses to losing in IMB

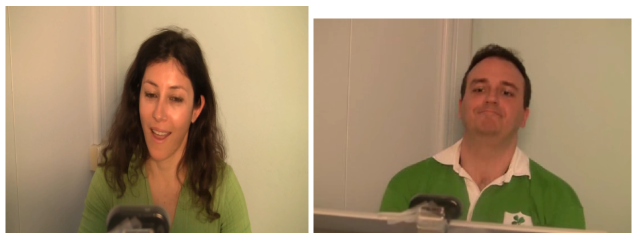

Fig. 10.2: Typical player responses to winning in IMB

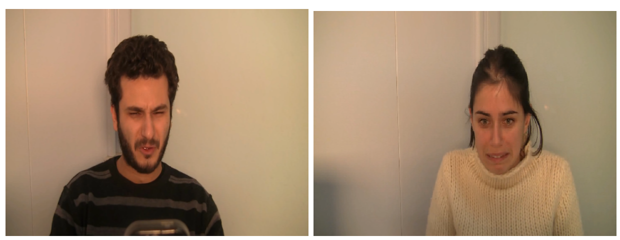

Fig. 10.3: Typical player responses to hard situations in IMB

number of occurrences of various events or the accumulated time spent doing a certain activity (such as the number of killings of a certain type of enemies or the total amount of time spent jumping). Features are also represented as sequences capturing the spatial and temporal order of events and allowing the discovery of temporal patterns [26]. Table 10.1 presents representative example features from each representation from.

Based on the features collected, a modelling approach is followed in an attempt to approximate the unknown function between game content, players' behaviour and how players experience the game. The player experience models are developed on different types and representations of features allowing a thorough analysis of the player-content relationship.

The following sections describe the approach followed to model player experience and the methodology proposed to tailor content generation for particular players using the constructed models as measures of content quality.

Table 10.1: The different types of representations of content and gameplay features.

Feature	Description
	Flat platform
( )(,)	A sequence of three coins
$(R^{\blacktriangleright}, R^{\blacktriangleright\Uparrow})$( )	Moving then jumping in the right direction when encountering an enemy
(,)( )	A gap followed by a decrease in platform height
$(\Uparrow^{\blacktriangleright})(S)(\blacktriangleright)$	Jumping to the right followed by standing still then moving right
t_{right}	Time spent moving right
n_{jump}	Total number of jumps
n_{coin}	Total number of coins
k_{stomp}	Number of enemies killed by stomping
N_e	Total number of enemies
B	Total number of blocks

10.5.1 Player experience modeling

When constructing player experience models, one should identify relevant features from game content and player behavior that affect player experience. This could be done by recording gameplay sessions and extracting several features as indicators of players' affect, performance and playing characteristics. Given the large size of the feature set that could be extracted, feature selection becomes a critical step for efficient knowledge discovery. The selection of the relevant subset of features not only helps us reduce the dimension of the input space resulting in more accurate models that are easier to analyse, but it also eliminates noisy features that are irrelevant for the player experience modelling as well as improving the model's generalisation capabilities.

Feature selection was applied as a first step when modelling players' experience. The input space constitutes of the different features extracted from the gameplay sessions. The models are trained to predict reported player experience from a subset of selected features. There are many approaches that could be followed to select the relevant subset of features, the feature selection method followed in this example is Sequential Forward Selection (SFS) combined with neuroevolutionary preference learning (using Single Layer Perceptrons (SLP) and simple Multi-Layer Perceptrons (MLPs) consisting of one layer of two hidden neurons) to measure the performance of each subset of features selected. Applying this approach yields different subsets of selected features for predicting each reported emotional states.

The underlying function between gameplay, content features and reported player experience is considered to be complex and cannot be easily captured using SLPs and simple MLPs and robust estimators are required if we are to accurately model the features-experience relationship. Therefore, once all features that contribute to accurate simple MLP models are found, an optimisation step is followed to build more powerful MLPs with more sophisticated structures. This is achieved by gradually increasing the complexity of the MLPs through adding hidden nodes and layers while monitoring the models' performance. Figure 10.4 presents an overview of the process followed to construct the PEMs.

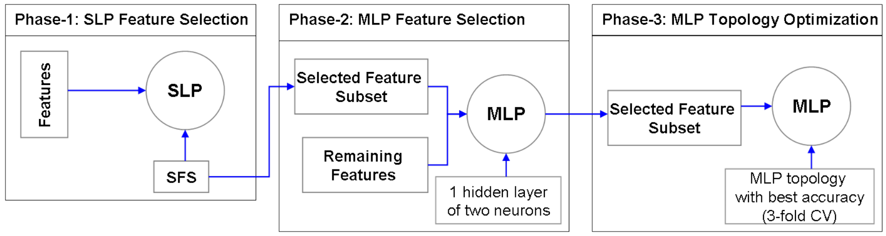

Fig. 10.4: The three-phase player experience modeling approach followed.

Following this approach, models with high accuracies were constructed for predicting players' reports of engagement, frustration and challenge from different subsets of features from different modalities. The models constructed were also of varying topologies and prediction accuracies.

10.5.2 Grammatical based personalised level generator

In Chapter **??**, we described how Grammatical Evolution (GE) can be used to evolve content for IMB. As discussed earlier, GE employs a design grammar to specify the structure of possible level designs. The grammar is used by GE to transform the phenotype into a level structure by specifying the types and properties of the different game elements that will be presented in the final level design. The fitness function defined in our earlier description was a purely aesthetics-based measure that score designs based on the number of elements presented and their placement properties.

In this section, we present a method for using the GE-based content generator to evolve personalised content for IMB. This is achieved by employing an adaptation mechanism as a fitness function to optimise player experience. The content is ranked according to the experience it evokes for a specific player and the content generator searches the resulting space for content that maximises particular aspects of player experience. To facilitate this, the player experience models constructed are used as fitness functions scoring each design evolved according to its appeal for a particular player. The fitness value assigned for each individual in the population (a level design) in the evolutionary process is the output of the player experience model which is the predicted value of an emotional state. The PEMs output is calculated by computing the values of the models's inputs; this includes the values of the content features which are directly calculated for each level design generated by GE and the values of the gameplay features estimated from the player's behavioural style while playing a test level.

The search for the best content features that optimise a particular state is guided by the model's prediction of the player experience states, with a higher fitness given

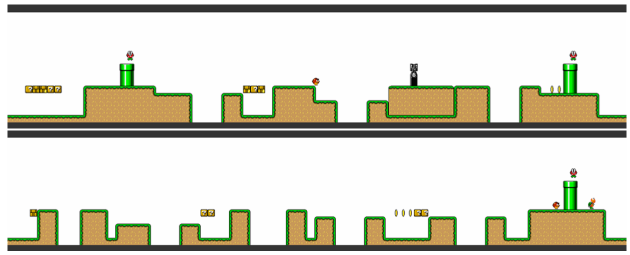

Fig. 10.5: The best levels evolved to maximise predicted challenge for two AI agents.

to the individuals that are predicted to be more engaging, frustrating or challenging for a particular player.

10.5.2.1 On-line personalised content generation

While the level is being played, the playing style is recorded and then used by GE to evaluate each individual design generated. Each individual is given a fitness according to the recorded player behaviour and the values of its content features. The best individual found by GE is then visualised for the player to play.

It is assumed that the player's playing style is largely maintained during consecutive game sessions and thus his playing characteristics in a previous level provide a reliable estimator of his gameplay behaviour in the next level. To compensate for the effect of learning while playing a series of levels, the adaptation mechanism only considers the recent playing style, i.e. the one which the player exhibited in the most recent level. Thus, in order to effectively study the behaviour of the adaptation mechanism, it is important to monitor this behaviour over time. For this purpose, AI agents with varying playing characteristics have been employed to test the adaptation mechanism since this requires the player to play-test a large number of levels. Figure 10.5 presents the best levels evolved to optimise player experience of challenge for two AI agents with different playing styles. The levels clearly exhibit different structures; a slightly more challenging level with more gaps was evolved for the second agent with more gaps and enemies than the one generated for the first agent.

10.6 Lab exercise: Generate personalised level for Super Mario Bors

In this lab session, you will generate personalised levels for a specific player using the InfiTux benchmark. The same software interface illustrated in Chapter 3 **??** will be used but this time you should focus on customisation of content for specific playing style so that the output of your *generateLevel* method should be player-driven content.

In order to facilitate meaningful detection of player experience and to allow you to develop player experience models, you will be given a dataset of 597 instances containing several statistical gameplay and content features collected from hundreds of players playing the game. The data contains information about several aspects of players' behaviour captured through features representing the frequencies of performing specific actions such as killing an enemy of jumping and the time spent doing certain behaviour such as moving right or jumping. Your task is to use this data to build PEM using a machine learning or a dataminig technique of your choice. The models you build can then be used to recognise the playing style of a new player.

After you build the models and successfully detect player experience, you should implement a method to adjust game content to change how player experience the game. You can adopt well-known concepts of player experience such as *fun*, *challenge*, *difficulty* or *frustration* and adjust the game content according to the aspect you would like your player to experience.

10.7 Summary

References

1. Bänziger, T., Tran, V., Scherer, K.R.: The geneva emotion wheel: A tool for the verbal report of emotional reactions. Poster presented at ISRE (2005)
2. Burelli, P.: Virtual cinematography in games: Investigating the impact on player experience. In: Proceedings of the International Conference on the Foundations of Digital Games, pp. 134–141 (2013)
3. Calleja, G.: In-Game: From immersion to incorporation. The MIT Press (2011)
4. Conati, C.: Probabilistic assessment of user's emotions in educational games. Applied Artificial Intelligence **16**(7-8), 555–575 (2002)
5. El-Nasr, M.S., Vasilakos, A., Rao, C., Zupko, J.: Dynamic intelligent lighting for directing visual attention in interactive 3-d scenes. Computational Intelligence and AI in Games, IEEE Transactions on **1**(2), 145–153 (2009)
6. Eladhari, M., Nieuwdorp, R., Fridenfalk, M.: The soundtrack of your mind: mind music-adaptive audio for game characters. In: Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology, p. 54. ACM (2006)
7. Giannatos, S., Nelson, M.J., Cheong, Y.G., Yannakakis, G.N.: Suggesting new plot elements for an interactive story. In: Intelligent Narrative Technologies (2011)
8. Gratch, J., Marsella, S.: A domain-independent framework for modeling emotion. Cognitive Systems Research **5**(4), 269–306 (2004)
9. Holmgård, C., Yannakakis, G.N., Karstoft, K.I., Andersen, H.S.: Stress detection for ptsd via the startlemart game. In: Affective Computing and Intelligent Interaction and Workshops, 2013. ACII 2013. 5th International Conference on. IEEE (2013)
10. IJsselsteijn, W., de Kort, Y., Poels, K., Jurgelionis, A., Bellotti, F.: Characterising and measuring user experiences in digital games. In: International Conference on Advances in Computer Entertainment Technology, vol. 2, p. 27 (2007)

11. Joachims, T.: Optimizing search engines using clickthrough data. In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 133–142. ACM (2002)
12. Johnson, L., Yannakakis, G.N., Togelius, J.: Cellular automata for real-time generation of infinite cave levels. In: Proceedings of the 2010 Workshop on Procedural Content Generation in Games, p. 10. ACM (2010)
13. Juul, J.: Games telling stories? a brief note on games and narratives. Game studies **1**(1), 40–62 (2001)
14. Martinez, H.P., Bengio, Y., Yannakakis, G.N.: Learning deep physiological models of affect. Computational Intelligence Magazine, IEEE **8**(2), 20–33 (2013)
15. Martínez, H.P., Garbarino, M., Yannakakis, G.N.: Generic physiological features as predictors of player experience. In: Affective Computing and Intelligent Interaction, pp. 267–276. Springer (2011)
16. Martínez, H.P., Yannakakis, G.N.: Mining multimodal sequential patterns: a case study on affect detection. In: Proceedings of the 13th international conference on multimodal interfaces, pp. 3–10. ACM (2011)
17. Metallinou, A., Narayanan, S.: Annotation and processing of continuous emotional attributes: Challenges and opportunities. In: 2nd International Workshop on Emotion Representation, Analysis and Synthesis in Continuous Time and Space (EmoSPACE 2013) (2013)
18. Nijholt, A.: Bci for games: A state of the artsurvey. In: Entertainment Computing-ICEC 2008, pp. 225–228. Springer (2009)
19. Nintendo Creative Department: (1985). Super Mario Bros, Nintendo
20. Persson, M.: Infinite mario bros. URL http://www.mojang.com/notch/mario/
21. Plans, D., Morelli, D.: Experience-driven procedural music generation for games
22. Savva, N., Scarinzi, A., Berthouze, N.: Continuous recognition of player's affective body expression as dynamic quality of aesthetic experience. Computational Intelligence and AI in Games, IEEE Transactions on (2012)
23. Shaker, N., Asteriadis, S., Yannakakis, G.N., Karpouzis, K.: A game-based corpus for analysing the interplay between game context and player experience. In: Affective Computing and Intelligent Interaction, pp. 547–556. Springer (2011)
24. Shaker, N., Asteridadis, S., Karpouzis, K., Yannakakis, G.N.: Fusing visual and behavioral cues for modeling user experience in games. IEEE Transactions on System Man and Cybernetics; Part B: Special Issue on Modern Control for Computer Games
25. Shaker, N., Togelius, J., Yannakakis, G.N.: Towards automatic personalized content generation for platform games. In: Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE) (2010)
26. Shaker, N., Yannakakis, G., Togelius, J.: Crowdsourcing the aesthetics of platform games. Computational Intelligence and AI in Games, IEEE Transactions on **5**(3), 276–290 (2013)
27. Togelius, J., Preuss, M., Beume, N., Wessing, S., Hagelbäck, J., Yannakakis, G.N., Grappiolo, C.: Controllable procedural map generation via multiobjective evolution. Genetic Programming and Evolvable Machines pp. 1–33 (2013)
28. Togelius, J., Schmidhuber, J.: An experiment in automatic game design. In: Computational Intelligence and Games, 2008. CIG'08. IEEE Symposium On, pp. 111–118. IEEE (2008)
29. Tognetti, S., Garbarino, M., Bonarini, A., Matteucci, M.: Modeling enjoyment preference from physiological responses in a car racing game. In: Computational Intelligence and Games (CIG), 2010 IEEE Symposium on, pp. 321–328. IEEE (2010)
30. Yannakakis, G.N.: Preference learning for affective modeling. In: Affective Computing and Intelligent Interaction and Workshops, 2009. ACII 2009. 3rd International Conference on, pp. 1–6. IEEE (2009)
31. Yannakakis, G.N., Hallam, J.: Ranking vs. preference: a comparative study of self-reporting. In: Affective Computing and Intelligent Interaction, pp. 437–446. Springer (2011)
32. Yannakakis, G.N., Martínez, H.P., Jhala, A.: Towards affective camera control in games. User Modeling and User-Adapted Interaction **20**(4), 313–340 (2010)
33. Yannakakis, G.N., Paiva, A.: Emotion in games. In: R.C. et al. (ed.) Handbook of Affective Computing. Oxford University Press

34. Yannakakis, G.N., Spronck, P., Loiacono, D., Andre, E.: Player modeling. In: Dagstuhl Seminar on Game Artificial and Computational Intelligence (2013)
35. Yannakakis, G.N., Togelius, J.: Experience-driven procedural content generation. Affective Computing, IEEE Transactions on **2**(3), 147–161 (2011)