Experience-Driven Procedural Content Generation (Extended Abstract)

(Invited Paper)

Georgios N. Yannakakis
Institute of Digital Games, University of Malta
Email: georgios.yannakakis@um.edu.mt

Julian Togelius
Dept. of Computer Science and Engineering, New York University
Email: julian@togelius.com

Abstract—**Procedural content generation is an increasingly important area of technology within modern human-computer interaction with direct applications in digital games, the semantic web, and interface, media and software design. The personalization of experience via the modeling of the user, coupled with the appropriate adjustment of the content according to user needs and preferences are important steps towards effective and meaningful content generation. This paper introduces a framework for procedural content generation driven by computational models of user experience we name *Experience-Driven Procedural Content Generation*. While the framework is generic and applicable to various subareas of human computer interaction, we employ games as an indicative example of content-intensive software that enables rich forms of interaction.**

***Keywords*—user experience; procedural content generation; games; affective loop**

I. Introduction

As information about users is becoming more readily available for all kinds of digital services and modern software development relies upon content creation, opportunity and demand for automatically generated personalized content increases in domains as diverse as e-commerce, news reading, web services, human-computer interfaces and digital games. When it comes to users of digital games (players), recent years have seen both a boost in the size of their population and their demographic diversification [1]. Twenty years ago, game players were largely young white males with an interest in technology; nowadays, gamers can be found in every part of society [2]. This means that skills, preferences and emotion elicitation differ widely among prospective players of the same game. Therefore, the need for tailoring the game to individual playing experience is growing and the tasks of user modeling and affective-based adaptation within games becomes increasingly difficult. Game engines that are able to recognize and model the playing style and detect the affective state of the user will be necessary milestones towards the personalization of the playing experience, as will procedural mechanisms that are able to adjust elements of the game to optimize for the experience of the player. In the following section we define the fundamental elements of the *Experience-Driven Procedural Content Generation* (EDPCG) framework [3] that realizes this vision.

A. Key Definitions

Computer games are dynamic media that implement rich forms of user interactivity. They also allow for high levels of player incorporation and yield dynamic and complex emotion manifestations. In our definition *player experience* is the collection of affective patterns elicited, cognitive processes emerged and behavioral traits observed during gameplay [4].

Game content, a central element of EDPCG, refers to all aspects of a game that affect the player experience but are not non-player character (NPC) behavior or the game engine itself. This definition includes such aspects as maps, levels, stories, quests, characters, rulesets, camera profiles, music, sound effects, and weapons. Content in games can be seen through the lens of five facets of human (or computational) creativity: game design, level architecture, visuals, audio, and narrative [5].

Procedural content generation (PCG) refers to the creation of game content — as defined above — automatically (or semi-automatically), through algorithmic means. When it comes to the development of a modern computer game, the effort and time required for the creation of game content represents a major part of the development cost (and time). Clearly, any technology that can alleviate the enormous burden of content creation and make it easier to tailor content to individual players or groups of players is warmly welcomed by game developers, game critics and the game-playing public in general. Attempts at generating game content procedurally have a fairly long history as PCG aspects appear in games such as *Rogue* (Toy and Wichman, 1980), *Diablo* (Blizzard Entertainment, 1996) and *Elite* (Acornsoft, 1984). Up until recently, PCG was almost only used in narrowly specialized roles and almost always during the development of the game; in the last few years, more and more games feature runtime PCG of some sort. While commercial games have been employing PCG algorithms since the 80s, academic research on PCG methods (such as the search-based PCG paradigm [6]) has bloomed only in the last five years. Beyond games, PCG research interacts with research areas such as human media interaction, computational aesthetics, computational creativity and recommender systems. In this paper however, we focus on and discuss PCG for games [7], and at the end of the paper we will return to how the ideas expressed here are applicable to other domains and research areas.

As games offer one of the most representative examples of rich and diverse content creation applications and are elicitors of unique user experiences we view game content as the *building block* of games and the generated games as the *potentiators* of player experience. Based on the above, the EDPCG framework [3] is defined as a generic and effective approach for the optimization of user (player) experience via the adaptation of the experienced content.

B. Realizing the Affective Loop

Affective computing [8], which is concerned with the study of emotions within human computer interaction, views the successful realization of the *affective loop* [9] as one of its ultimate goals. Games are, by nature, interactive entertaining activities that are played within virtual worlds. Therefore any potential limitations of affective interaction — such as the justifiability of affect-based, automated, game decisions — are eliminated [4]. Games are designed to offer affective experiences influenced by player feedback and many games let players go through frustrating, anxious, or fearful episodes of play in order to heighten involvement. Therefore a user under gaming conditions — more than any other form of human computer interaction — is generally open to affect-based alterations of the interaction, and this influences the game experience directly. Given all the above properties of games we argue that they can offer the best and most meaningful realization of the affective loop [4].

To successfully close the affective loop [9] within games one needs to fulfill a set of system requirements: the game should be tailored to individual players' affective response patterns; the game adaptation should be fast, yet not necessarily noticeable; and the affect-based interaction should be rich in terms of game context, adjustable game elements and player input. The EDPCG framework satisfies these conditions via the efficient generation of game content which is driven by models of player experience. Thus we argue that EDPCG offers a holistic realization of affective interaction as it elicits emotion through variant game content types, it integrates game content to computational models of user affect and, finally, it utilizes game content for the adaptation of the experience [4].

C. This Paper

Since its introduction in [3], the EDPCG framework has become one of the core research trends in the study of procedural content generation, computational game creativity and game artificial intelligence. While EDPCG is a framework with foundations in game technology and affective computing it has managed to penetrate adjacent areas of research within human computer interaction [10]. This paper provides an extended synopsis of the EDPCG framework as introduced in [3], it highlights its influence to variant research areas, and discusses future applications of the framework. The interested reader may refer to [3] for a comprehensive literature review.

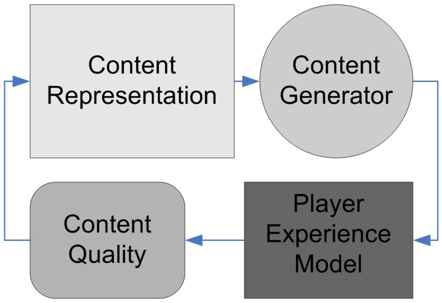


Fig. 1. The four key components of the EDPCG framework.

II. The EDPCG Framework

EDPCG defines a novel approach to PCG coupling player experience with procedural content generation. Since games are composed by game content that, when played by a particular player, elicit experience patterns, one needs to assess the quality of the content generated (linked to the experience of the player), search through the available content, and generate content that optimizes the experience for the player (see Fig. 1). In particular, the key components of EDPCG are:

- **Player experience modeling**: player experience is modeled as a function of game content and player.
- **Content quality**: the quality of the generated content is assessed and linked to the modeled experience.
- **Content representation**: content is represented accordingly to maximize search efficacy and robustness.
- **Content generator**: the generator searches through content space for content that optimizes the experience for the player according to the acquired model.

Below, we survey the four main components of EDPCG, provide a taxonomy of different approaches to each and outline the main research challenges faced. Each component of EDPCG has its own dedicated literature and the extensive review of each is beyond the scope of this paper. A detailed survey is available in [3].

A. Player Experience Modeling

Player experience models can be built on different types of data collected from players which in turn define different approaches to player experience modeling (PEM). We can identify three main classes of methods for modeling player experience in games which rely on 1) data expressed by players (*subjective* PEM); 2) data obtained from alternative modalities of player response such as physical measurement (*objective* PEM); and 3) data obtained through the interaction between the player and the game (*gameplay-based* PEM). The PEM approaches can be combined to more powerful hybrid methods for capturing player experience. We discuss the three approaches below.

1) Subjective PEM: The most direct way to develop a model of experience is to ask the players themselves about their playing experience and build a model based on the obtained data. Subjective PEM considers only first person reports (self-reports) and not reports expressed indirectly by experts or external observers. Subjective player experience modeling can be based on either players' *free-response* during play or on *forced* data retrieved through questionnaires. Forced self-reports can be classified as *rating*, in which the players are asked to answer questionnaire items given in a rating form [11][1], and *preference*, in which players are asked to compare and rank their experience in two or more variants/sessions of the game [12], [13].

Subjective PEM may yield very accurate models of user states [12]; however, there are well known inherent limitations of this approach. First, there are usually significant subjectivity biases in the responses of players which may be caused by player learning and self-deception effects (among many other factors). Second, self-reports can be intrusive if questionnaire items are presented during the gameplay sessions [14] or sensitive to memory effects if players are asked to express their experience after a lengthy game session.

2) Objective PEM: Player experience can be linked to a set of emotions, which may be active simultaneously, usually triggered by events occurring during gameplay. Games can elicit player emotional responses which in turn 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 may assist in recognizing and synthesizing the emotional responses of the player. The *objective* approach to PEM incorporates access to multiple modalities of player input for the purpose of modeling the affective state of the player during play.

Within objective PEM, a number of real-time recordings of the player may be investigated for modeling affective aspects of player experience. There are several studies that explore the interplay between physiology and gameplay by investigating the impact of different gameplay stimuli to a number of dissimilar physiological signals. Such signals are obtained through electrocardiography (ECG) [13], galvanic skin response (GSR) [11], respiration [13], electroencephalography (EEG) [15], and electromyography (EMG).

In addition to physiology, one may track the player's bodily expressions (motion tracking) at different levels of detail and infer the real-time affective responses from the gameplay stimuli. Motion tracking may include body and head pose as well as gaze [16] and facial expression [17]. Speech may also be used for inferring player affective responses [18] but it is not directly applicable for the vast majority of game genres.

Implementations of the objective PEM approach can be *model-based* (top-down) or *model-free* (bottom-up). Model-based refers to models derived from emotion theories such as the popular emotional dimensions of arousal and valence [19] in which bodily responses are mapped to specific emotional responses — e.g. increased heart rate of a player corresponds to high arousal and player excitement. Model-free PEM refers to the construction of an unknown mapping between modalities of player input and an emotional state representation via annotated data. This approach is very common for the recognition of affect through annotated facial expressions.

Note that the space between a completely model-based and a completely model-free approach is a continuum, and any objective PEM approach might be placed somewhere along this axis. On that basis, all objective PEM approaches may be viewed as hybrids between the two ends of the spectrum, containing elements of both approaches.

The key limitations of the objective PEM approach include its high intrusiveness, low practicality (combined with high complexity) and questionable feasibility. Several modalities of player input are still implausible within commercial game development. For instance, existing techniques for physiological recording require physical contact between body parts (e.g., head, fingertips) and sensors, making physiological signals such as EEG and skin conductance impractical and intrusive. Modalities such as facial expression and speech could be technically plausible in games, even though most of the vision-based affect-detection systems currently available cannot operate in real time [20].

On a positive note, recent advances in sensor technology have resulted in low-cost unobtrusive biofeedback devices appropriate for gaming applications (such as the Emotiv[2] EEG system and Empatica's *Embrace* bracelet[3]). Advances in gaming peripherals, such as the Kinect motion controller, have already shown their great potential for multimodal natural interaction in games. Further, the Intel RealSense 3D camera (which is currently included as a standard device in many laptops and tablets) is already used in upcoming affect-intense games such as the *Nevermind* (Reynolds) horror game. Finally, top game developers such as *Valve* have already experimented with multiple modalities of player input (e.g. physiological and behavioral patterns) for the personalization of popular AAA games such as *Left 4 Dead* (Valve, 2008) [21].

3) Gameplay-based PEM: The main assumption that drives *gameplay-based* PEM is that player actions and real-time preferences are linked to player experience since games may affect the player's cognitive processing patterns and cognitive focus. Cognitive processes may, in turn, influence emotions; one may infer the player's emotional state by analyzing patterns of the interaction and associating user emotions with context variables [22]. Any element derived from the interaction between the player and the game forms the basis for gameplay-based PEM. This includes parameters from the player's behavior derived from responses to system elements (e.g. non-player characters or game levels).

As in objective PEM, a gameplay-based PEM approach can be classified as *model-based*, *model-free* or some hybrid between the two. Model-based approaches are typically inspired by a general theoretical framework of behavioral analysis and/or cognitive modeling (e.g. belief-desire-intention model, the OCC model [23], or Skinner's model [24]), but there are also theories about user affect that are specific to games

[1] In this paper we view class-based annotation as a form of binary rating.

[2] http://emotiv.com

[3] http://www.empatica.com

such as Malone's design components for fun games [25] and Koster's *theory of fun* [26].

Model-free approaches usually involve the processing and mining of the massive sets of player data (metrics) which modern games often collect [27]. While such data usually contains behavioral aspects of playing experience, data mining and data analysis research needs to focus on inferring the relationship between detailed player metrics and cognitive and affective maps of experience. The inputs to a gameplay-based player experience model are statistical spatio-temporal features of game interaction. Those features can be mapped to levels of cognitive states such as attention, challenge and engagement. General measures such as performance and time spent on a task have been used in the literature, but also game-specific measures such as the weapons selected in a shooter game or the times a player dies. Moreover, several different difficulty and challenge measures have been proposed for different game genres (see [28], [29] among many). Finally, a player profile or a player model [30] can be embedded in the process of gameplay-based PEM.

Gameplay-based PEM is certainly the most computationally efficient and least intrusive PEM approach of all three but it usually results in a low-resolution model of playing experience and its affective component. The models are often based on several strong assumptions that relate player experience to gameplay actions and preferences.

4) General Modeling Principles: A model of player experience predicts some aspect of the experience of a player in general, a type of player or a particular player would have in some game situation. As already mentioned, there are many ways this can be done, with approaches to player experience modeling varying both regarding the inputs (from what the experience is predicted, e.g. physiology, level design parameters, playing style or game speed), outputs (what sort of experience is predicted, e.g. fun, frustration, attention or immersion) and the modeling methodology.

If data recorded includes a scalar representation of experience, or classes and annotated labels of user states, using the PEM methods discussed above, any of a large number of machine learning (regression and classification) algorithms can be used to build affective models. On the other hand, if the ground truth of experience is given in a pairwise preference (rank) format (e.g. game version X is more frustrating than game version Y) standard supervised learning techniques are inapplicable, as the problem becomes one of *preference learning* [31], [12]. Available preference learning approaches include decision trees, artificial neural networks (shallow and deep architectures) and support vector machines [32].

B. Content Quality

In EDPCG, the main use of the acquired player models is to judge the quality (usefulness, fitness) of game content items. Assessing the quality of the content is necessary in the content generation phase, when candidate content items are evaluated and used to generate new content. The task of the evaluation function is to evaluate an item of game content and assign it a scalar (or a vector of real numbers) that accurately reflects its suitability for use in the game, and its capacity for instilling the desired experience. To do this, the evaluation function uses the PEM in some capacity. Designing the evaluation function is an ill-posed problem; the designer first needs to decide what, exactly, should be optimized and then how to formalize it. Perhaps the task of the evaluation function is to judge how entertaining some content would be for a given player, or how frustrating, how challenging etc.

Three key classes of evaluation functions can be distinguished for assessing the quality of generated content: *direct*, *simulation-based* and *interactive* functions.

1) Direct Evaluation Functions: In a direct evaluation function, some features are extracted from the generated content, and these features are mapped directly to a content quality value. Hypothetical such features might include the number of paths to the exit in a maze, the firing rate of a weapon, the spatial concentration of resources on a strategy map, and the material balance in randomly selected legal positions for board game rule set. The mapping between features and content quality might be linear or non-linear, but typically does not involve large amounts of computation, and is often specifically tailored to the particular game and content type. This mapping can be contingent on a model of the playing style, preferences or affective state of the player yielding an element of personalization for content generation. An important distinction within direct evaluation functions is between *theory-driven* and *data-driven* functions. In theory-driven functions, the designer is guided by intuition and/or some qualitative theory of emotion or player experience to derive a mapping between an experience model and the quality of content [33], [34], [35], [36]. On the other hand, data-driven functions are based on collecting data on the effect of various examples of content via e.g. questionnaires and/or physiological measurements and then using automated means to tune the mapping from content to player experience and finally to evaluation functions [37], [38], [39].

2) Simulation-based Evaluation Functions: It is not always apparent how to design a meaningful direct evaluation function for some game content — in many cases, it seems that the content must be interacted with for particular emotional responses to be elicited and evaluated. A simulation-based evaluation function is based on an artificial agent playing through some part of the game that involves the content being evaluated. Such playthrough might include finding the way out of a maze while not being killed or playing the board game that results from the newly generated rule set against another artificial agent. Features, that map to player experience models, are then extracted from the observed gameplay and used to calculate the quality value of the content.

A key distinction is between *static* and *dynamic* simulation-based functions. In a static evaluation function, it is not assumed that the agent changes while playing the game (e.g. in [40]); in a dynamic evaluation function the agent changes during the game and the quality value somehow incorporates this change (e.g. in [36], [41]).

It should be noted that while simulations of the game environment can typically be executed faster than real-time, simulation-based evaluation functions are in general more computationally expensive than direct evaluation functions; dynamic simulation-based evaluation functions can be time-consuming, all but ruling out online content generation.

3) Interactive Evaluation Functions: Interactive evaluation functions score content based on interaction with a player in the game, which means that fitness is evaluated during the actual gameplay. Data can be collected from the player either *explicitly*, using questionnaires or verbal input data, or *implicitly* by measuring e.g. how often or long a player chooses to interact with a particular piece of content [42], when the player quits the game, or expressions of affect such as intensity of button-presses, shaking the controller, physiological response, gaze fixation, speech quality, facial expressions and postures. Data is used to tailor the player experience models to the specific player, which in turn affects the evaluation function of the content presented to the player. If an interactive evaluation function is coupled with a subjective PEM component (e.g. questionnaire pop up during gameplay) the function is classified as *explicit*; otherwise (if coupled with objective or gameplay-based PEM) the function is classified as *implicit*. As mentioned earlier, the problem with explicit data collection is that it can interrupt the game play, whereas the problem with implicit data collection is that data may often be noisy, inaccurate, delayed and of low-resolution.

C. Content Representation

A central question in EDPCG concerns how to represent whatever is generated. Content may be represented symbolically within a tree or a graph data structure. That is usually the practice in the computational narrative community (see e.g. [43]). While symbolic representation can be human-readable and human-editable, non-symbolic representations such as artificial neural networks might allow for more effective search in many domains. EDPCG primarily focuses on bottom-up, search-based [6] approaches for generating content, which are driven by computational heuristics of player experience. As the most common search-based PCG methods use evolutionary algorithms, an important question is how genotypes (i.e. the data structures that are internally represented by the content generator) are mapped to phenotypes (i.e. the data structure that is assessed by the evaluation function). An important distinction among representations is between *direct encodings*, wherein the size of the genotype is linearly proportional to the size of phenotype and each part of the genome maps to a specific part of the phenotype, and *indirect encodings*, wherein the genotype maps nonlinearly to the genotype and the former need not be proportional to the latter.

As a concrete example, a level for a 2D platform game (such as *Super Mario Bros*) might be represented: 1) *directly* as a two-dimensional grid where the contents of each cell (e.g. ground, coin, wall, enemy, free space) is specified separately, and mutation works by changing directly on the cells; 2) *more indirectly* as a list of positions and shapes of walls and pieces of ground that each occupy more than a single cell in the underlying grid, and another list of positions of enemies and items; 3) *even more indirectly* as a repository of different reusable patterns of walls and free space (e.g. a long jump followed by a particular type of enemy), and a list of how they are distributed across the level; 4) *very indirectly* as a list of desirable properties (e.g. number of gaps, distribution of gaps, number of enemies, average height of coins over ground); or 5) *most indirectly* as a random number seed.

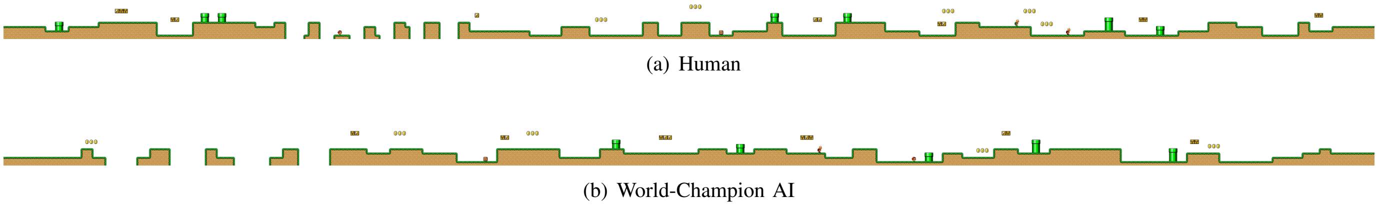
D. Content Generator

Once player experience is captured, content is appropriately represented and content evaluation functions are designed, the content generator needs to search within the resulting search space for content that maximizes particular aspects of player experience. If content is represented via a small number of dimensions (indirectly) exhaustive search should be able to provide robust solutions for online PCG [44]. In general, the more direct the representation is the larger the content search space becomes. Where exhaustive search is infeasible, other techniques could be used varying from simple heuristic and gradient-search (if gradient is computable) [45] to stochastic global optimization techniques such as evolutionary algorithms and particle swarm optimization [6]. Ideally, the content generator should be able to identify *if*, *how much* and *how often* content should be generated for a particular player.

III. EDPCG in Practice: Personalized Level Creation in Super Mario Bros

Herein we give the reader a feel for what EDPCG entails by providing a popular example from recently published papers. We take our example from Pedersen et al. [39], who modified an open-source clone of the classic platform game *Super Mario Bros* (Nintendo, 1985) to allow for personalized level generation. The realization of EDPCG in this example is illustrated in Fig. 3.

The first step was to represent the levels in a format that would yield an easily searchable space. A level was represented as a short parameter vector describing the number, size and placement of gaps which the player can fall through, and the presence or absence of a switching mechanic. The next step was to create a model of player experience based on the level played and the player's playing style. Data was collected from hundreds of players, who played pairs of levels with different parameters and were asked to rank which of two levels best induced each of the following user states: fun, challenge, frustration, predictability, anxiety, boredom. While playing, the game also recorded a number of metrics of the players' playing styles, such as the frequency of jumping, running and shooting. This data was then used to train neural networks to predict the examined user states using evolutionary preference learning. Finally, these models were used to optimize game levels for particular players [44]. Two examples of such levels can be seen in Fig. 2.



(a) Human

(b) World-Champion AI

Fig. 2. Example levels generated for two different Super Mario players. The generated levels maximize the modeled *fun* value for each player. The level on top is generated for one of the experiment subjects that participated in [39] while the level below is generated for the world champion agent of the Mario AI competition.

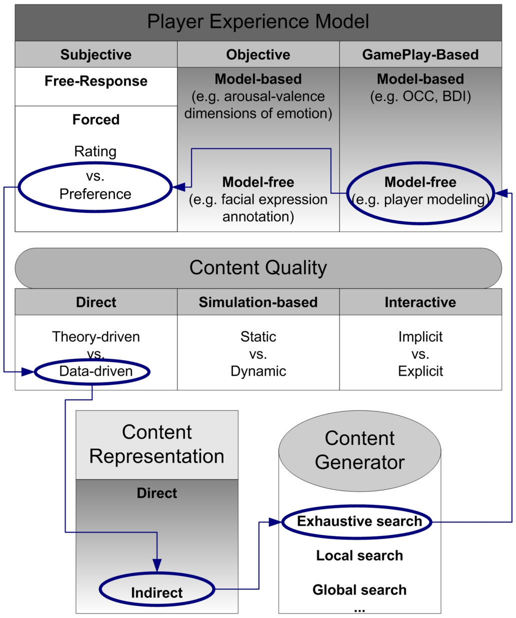


Fig. 3. The EDPCG framework in detail. The gradient grayscale-colored boxes represent a continuum of possibilities between the two ends of the box while white boxes represent discrete, exclusive, options within the box. The blue arrows illustrate the EDPCG approach followed for the Super Mario Bros example study [39], [44]: Content quality is assessed via a direct, data-driven evaluation function which is based on a combination of a gameplay-based (model-free), and a subjective (pairwise preference) player experience modeling approach; content is represented indirectly and exhaustive search is applied to generate better content.

IV. The Road Ahead

Almost 4 years after the introduction of EDPCG [3] the framework constitutes a core research trend within procedural content generation and has also influenced the larger areas of affective interaction and game artificial intelligence. The use of procedural content generation techniques for the design of better games has reached a frenzy of interest in commercial and indie game development which is showcased by successful (almost entirely procedurally generated) games such as *Minecraft* (Mojang, 2011) and *Spelunky* (Mossmouth, 2009), and the hype surrounding upcoming games such as *No Man's Sky* (Hello Games). Future games are, in general, expected to contain less manual and more user- or procedurally generated content, as the cost of content creation and the resulted content creation bottleneck are key challenges for commercial game production. As the number of games that are partially or fully automatically generated grows, the challenge of detecting and monitoring player experience in never-ending open worlds of infinite replayability value increases substantially. The automation of content creation, however, offers a unique opportunity toward realizing affect-driven content generation in games.

Arguably, the need of automatic personalized content generation expands beyond games. While EDPCG is inspired by and built for games its applicability to other human computer interaction domains is rather obvious. Recommender systems, semantic web applications, intelligent tutoring systems, interface design and computational creativity and art are some of the diverse sub-domains EDPCG has already being suitable for. We can imagine such "content" as personalized exercise plans, furniture assembly instructions, decorative elements (for use as Windows backgrounds or printed on 3D printers and placed on the window porch), schedules, menu systems and shopping lists to be generated via non-game EDPCG.

Even though EDPCG has influenced the above research fields at large there are several interesting research questions that require much more study. These include the appropriate representation of game content and the design of relevant and computationally efficient evaluation functions based on reliable computational models of player experience. The potential gains from providing good solutions to these challenges are significant: the invention of new game genres built on PCG, streamlining of the game development process, and further understanding of the mechanisms of human entertainment, creativity and player emotion are all possible.

It is important to note that EDPCG constitutes an innovative mixture of three initiatives: data-driven (through search), user-driven (through player experience modeling) and designer-driven (through parameter design) content creation. In EDPCG, the user drives the generation of new (personalized) content. The designer's role becomes that of making high-level decisions about the type of content to be generated and the type of experience to be optimized. EDPCG moves the designer's role up the value chain while saving labor and, as a result, extends the limits of what technology can do.

Acknowledgment

The research was supported, in part, by the FP7 ICT projects C2Learn (318480) and iLearnRW (318803).

References

[1] J. Juul, *A Casual Revolution: Reinventing Video Games and Their Players*. MIT Press, 2009.

[2] T. Taylor, *Play Between Worlds: Exploring Online Game Culture*. The MIT Press, March 2006.

[3] G. N. Yannakakis and J. Togelius, "Experience-driven procedural content generation," *Affective Computing, IEEE Transactions on*, vol. 2, no. 3, pp. 147–161, 2011.

[4] G. N. Yannakakis and A. Paiva, in *The Oxford Handbook on Affective Computing*, R. Calvo, S. D'Mello, J. Gratch, and A. Kappas, Eds., 2015, ch. Emotion in games.

[5] A. Liapis, G. N. Yannakakis, and J. Togelius, "Computational game creativity," in *Proceedings of the Fifth International Conference on Computational Creativity*, 2014, pp. 285–292.

[6] 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.

[7] N. Shaker, J. Togelius, and M. J. Nelson, *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*. Springer, 2015.

[8] R. W. Picard, *Affective Computing*. Cambridge, MA: The MIT Press, 1997.

[9] P. Sundström, "Exploring the affective loop," Stockholm University, Tech. Rep., 2005.

[10] G. N. Yannakakis and J. Togelius, "A panorama of artificial and computational intelligence in games," *Computational Intelligence and AI in Games, IEEE Transactions on*, 2014.

[11] R. L. Mandryk, K. M. Inkpen, and T. W. Calvert, "Using Psychophysiological Techniques to Measure User Experience with Entertainment Technologies," *Behaviour and Information Technology (Special Issue on User Experience)*, vol. 25, no. 2, pp. 141–158, 2006.

[12] G. N. Yannakakis, "Preference Learning for Affective Modeling," in *Proceedings of the Int. Conf. on Affective Computing and Intelligent Interaction*. Amsterdam, The Netherlands: IEEE, September 2009, pp. 126–131.

[13] S. Tognetti, M. Garbarino, A. Bonarini, and M. Matteucci, "Modeling enjoyment preference from physiological responses in a car racing game," in *Proceedings of IEEE CIG*, 2010, pp. 321–328.

[14] R. J. Pagulayan, K. Keeker, D. Wixon, R. L. Romero, and T. Fuller, *User-centered design in games. The HCI Handbook*. Lawrence Erlbaum Associates, 2003.

[15] A. Nijholt, "BCI for Games: A State of the Art Survey," in *Proceedings of Entertainment Computing - ICEC 2008*, 2009, pp. 225–228.

[16] N. Shaker, S. Asteriadis, G. N. Yannakakis, and K. Karpouzis, "Fusing visual and behavioral cues for modeling user experience in games," *Cybernetics, IEEE Transactions on*, vol. 43, no. 6, pp. 1519–1531, 2013.

[17] M. Pantic and G. Caridakis, *Emotion-Oriented Systems: The Humaine Handbook*. Springer-Verlag Berlin Heidelberg, 2011, ch. Image and Video Processing for Affective Applications, pp. 101–117.

[18] J. Kim and E. Andre, "Emotion recognition using physiological and speech signal in short-term observation," in *International Tutorial and Research Workshop on Perception and Interactive Technologies*. Germany: Springer-Verlag Berlin Heidelberg, 2006, pp. 53–64.

[19] J. A. Russell, "Core affect and the psychological construction of emotion," *Psychological Rev.*, vol. 110, pp. 145–172, 2003.

[20] Z. Zeng, M. Pantic, G. Roisman, and T. Huang, "A survey of affect recognition methods: Audio, visual, and spontaneous expressions," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 31, no. 1, pp. 39–58, 2009.

[21] M. Ambinder, "Biofeedback in gameplay: How valve measures physiology to enhance gaming experience," in *Game Developers Conference*, San Francisco, California, US, 2011.

[22] J. Gratch and S. Marsella, "Evaluating a computational model of emotion," *Autonomous Agents and Multi-Agent Systems*, vol. 11, no. 1, pp. 23–43, 2005.

[23] A. Ortony, G. L. Clore, and A. Collins, *The Cognitive Structure of Emotions*. Cambridge University Press, 1988.

[24] B. F. Skinner, *The Behavior of Organisms: An Experimental Analysis*. Cambridge, Massachusetts: B. F. Skinner Foundation, 1938.

[25] T. W. Malone, "What makes things fun to learn? heuristics for designing instructional computer games," in *Proceedings of the 3rd ACM SIGSMALL symposium and the first SIGPC symposium on Small systems*, 1980, pp. 162–169.

[26] R. Koster, *A theory of fun for game design*. Paraglyph press, 2005.

[27] A. Drachen, C. Thurau, J. Togelius, G. N. Yannakakis, and C. Bauckhage, "Game data mining," in *Game Analytics*. Springer, 2013, pp. 205–253.

[28] G. van Lankveld, P. Spronck, and M. Rauterberg, "Difficulty Scaling through Incongruity," in *Proceedings of the 4th International Artificial Intelligence and Interactive Digital Entertainment Conference*. AAAI Press, 2008, pp. 228–229.

[29] G. Andrade, G. Ramalho, H. Santana, and V. Corruble, "Extending reinforcement learning to provide dynamic game balancing," in *Proceedings of the Workshop on Reasoning, Representation, and Learning in Computer Games, IJCAI*, August 2005, pp. 7–12.

[30] G. N. Yannakakis, P. Spronck, D. Loiacono, and E. André, "Player modeling," *Artificial and Computational Intelligence in Games*, vol. 6, pp. 45–59, 2013.

[31] J. Fürnkranz and E. Hüllermeier, "Preference learning," *Künstliche Intelligenz*, vol. 19, no. 1, pp. 60–61, 2005.

[32] V. E. Farrugia, H. P. Martínez, and G. N. Yannakakis, "The preference learning toolbox," *arXiv preprint arXiv:1506.01709*, 2015.

[33] Y. Cheong and M. Young, "A computational model of narrative generation for suspense," in *AAAI 2006 Computational Aesthetic Workshop*, 2006.

[34] N. Sorenson and P. Pasquier, "Towards a generic framework for automated video game level creation," in *Proceedings of the European Conference on Applications of Evolutionary Computation (EvoApplications)*, vol. 6024. Springer LNCS, 2010, pp. 130–139.

[35] J. Togelius, M. Preuss, and G. N. Yannakakis, "Towards multiobjective procedural map generation," in *Proceedings of the Workshop on Procedural Content Generation, Foundations of Digital Games*. Monterey, CA: ACM Press, June 2010.

[36] J. Togelius and J. Schmidhuber, "An Experiment in Automatic Game Design," in *Proceedings of the IEEE Symposium on Computational Intelligence and Games*. Perth, Australia: IEEE, December 2008, pp. 252–259.

[37] A. Drachen, A. Canossa, and G. N. Yannakakis, "Player Modeling using Self-Organization in Tomb Raider: Underworld," in *Proceedings of the IEEE Symposium on Computational Intelligence and Games*. Milan, Italy: IEEE, September 2009, pp. 1–8.

[38] O. Missura and T. Gärtner, "Player modeling for intelligent difficulty adjustment," in *Proceedings of the ECML–09 Workshop From Local Patterns to Global Models (LeGo–09)*, J. F. Arno Knobbe, Ed., Bled, Slovenia, September 2009.

[39] C. Pedersen, J. Togelius, and G. N. Yannakakis, "Modeling Player Experience for Content Creation," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 2, no. 1, pp. 54–67, 2010.

[40] J. Togelius, R. De Nardi, and S. M. Lucas, "Towards automatic personalised content creation in racing games," in *Proceedings of the IEEE Symposium on Computational Intelligence and Games*, 2007.

[41] C. Browne, "Automatic generation and evaluation of recombination games," Ph.D. dissertation, Queensland University of Technology, 2008.

[42] E. Hastings, R. Guha, and K. O. Stanley, "Evolving content in the galactic arms race video game," in *Proceedings of the IEEE Symposium on Computational Intelligence and Games*, 2009, pp. 241–248.

[43] M. O. Riedl and N. Sugandh, "Story planning with vignettes: Toward overcoming the content production bottleneck," in *Proceedings of the 1st Joint International Conference on Interactive Digital Storytelling*, Erfurt, Germany, 2008, pp. 168–179.

[44] N. Shaker, G. N. Yannakakis, and J. Togelius, "Towards Automatic Personalized Content Generation for Platform Games," in *Proceedings of AIIDE*. Palo Alto, CA: AAAI Press, October 2010, pp. 63–68.

[45] G. N. Yannakakis and J. Hallam, "Real-time Game Adaptation for Optimizing Player Satisfaction," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 1, no. 2, pp. 121–133, June 2009.