

Emotion-driven Level Generation

Julian Togelius and Georgios N. Yannakakis

Abstract This chapter examines the relationship between emotions and level generation. Grounded in the experience-driven procedural content generation framework we focus on levels and introduce a taxonomy of approaches for emotion-driven level generation. We then review four characteristic level generators of our earlier work that exemplify each one of the approaches introduced. We conclude the chapter with our vision on the future of emotion-driven level generation.

1 Introduction

Game levels are frequently capable of, and indeed designed to, elicit affective responses. Such responses range from the sadness of traversing a desolate landscape, to the feeling of achievement upon clearing a hard but fair challenge, to the delight of finding a hidden treasure cache, to the frustration of butting ones head against an abusively hard challenge, to the tense dread of exploring a dark maze where a monster might appear any second. The player might experience different and changing emotions while playing a single level. The affective response of players to games are influenced by numerous factors — many of them detailed in this book, such as sound effects, narrative and cinematography — but this particular chapter will focus on level design. We will be looking at level design as the arrangement of components or items from a given vocabulary in order to yield a space for the player character(s) to progress through. This can be exemplified by the multitude of levels designed in *Super Mario Maker* (Nintendo, 2015), levels which share a common and somewhat restricted set of items and affordances, but which explore a remarkably large expressive range and give rise to a wide variety of player emotions.

Julian Togelius

Dept. of Computer Science and Engineering, New York University e-mail: julian@togelius.com

Georgios N. Yannakakis

Institute of Digital Games, University of Malta e-mail: georgios.yannakakis@um.edu.mt

The structure of this paper will build on our own *experience-driven procedural content generation* framework, which describes how Procedural Content Generation (PCG) methods can be used to adapt games according to models of player experience [23]. From this perspective, 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. The potential that games have to influence players is mainly due to the rich contextual building blocks (i.e., game content) they offer and their ability of placing the player in a continuous mode of interaction with the game. Players are continuously presented with (and react to) a wide palette of content types that vary from sound effects and textures to narratives, game rules and levels. This rich interactivity can naturally accommodate mechanisms for real-time adaptation of game content aimed at adjusting player experience and realizing affective interaction [23].

In the rest of this chapter we provide a brief taxonomy of approaches for emotion-driven level generation by putting an emphasis on two core dimensions that influence the relationship between level generation and affect modeling. First, we distinguish between level generation that follows a player-centric approach and that which follows a designer-centric approach. Second, we differentiate between level generation approaches that consider affect directly and approaches that are built on other aspects of player experience such as cognitive patterns and player behaviors. We provide an example for each of the four possibilities. The chapter concludes with a vision of the future of emotion-driven level generation.

2 Emotion-driven Level Generation

Emotion-driven level generation can be viewed as an instance of the *experience-driven procedural content generation* framework [23]. According to our definitions in [23] *player experience* is the collection of affective patterns elicited, cognitive processes emerged and behavioral traits observed during gameplay [22]. *Game content* 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 game design, level architecture, visuals, audio, and narrative [9]. *Procedural content generation* (PCG) refers to the creation of game content — as defined above — automatically (or semi-automatically), through algorithmic means. 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 experience-driven PCG framework [23] is defined as a generic approach for the optimization of player experience via the adaptation of the experienced content.

To realize experience-driven PCG for level generation one needs to assess the quality of the level generated (linked to the experience of the user), search through the available level content, and generate a level that optimizes the experience for the

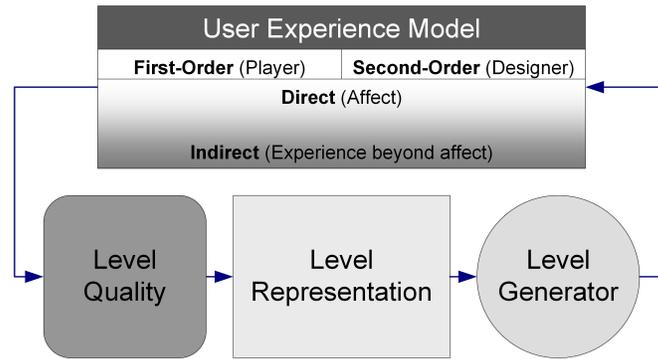


Fig. 1: The four key components of the experience-driven PCG framework [23] for level generation. The four user experience modeling options are detailed in the taxonomy of section 3.

user (see Figure 1). In particular, the key components of experience-driven PCG for level generation are:

- **User experience model:** user experience is modeled as a function of game content and the user. The user considered can be either a player (i.e., first-order level generation) or a designer (i.e., second-order level generation); see section 3.1 for further details. The modeling approach can be either direct or indirect depending on its level of grounding in user affect; see section 3.2 for further details. Section 3 introduces the taxonomy of the four aforementioned user experience modeling options for level generation, thereby, enriching the experience-driven PCG framework.
- **Level quality evaluator:** the quality of the generated content (i.e., level) is assessed and linked to the modeled experience.
- **Level representation:** the level is represented accordingly to maximize search efficacy and robustness.
- **Level generator:** the generator searches through content (i.e., parameterized level) space for content that optimizes the experience for the user according to the acquired model.

With respect to the user experience component of the experience-driven PCG framework, emotion-driven level design focuses on emotion and affect and takes into consideration other aspects of experience only implicitly (as discussed thoroughly in the following sections). With regards to the PCG component of experience-driven PCG, emotion-driven level generation considers game levels and their core architectural properties (functionality and aesthetics) as the content type under consideration. In other words emotion-driven level generation investigates the generation of game levels and their impact on gameplay and experience.

3 A Taxonomy of Emotion-driven Level Generators

According to the taxonomy presented in [19, 23] game content can be *necessary* (e.g. game rules or a main quest) or *optional* (e.g. trees in a level, flying birds on the background or a side quest). Necessary content needs to be completable or playable by the player, and generators of necessary content therefore needs to assure the completeness of the generated artefacts. We here consider levels to be necessary content for a digital game as most game levels need to be completable. Further, a generator can be either *offline* or *online*, *random* or based on a *parametrizable*, *stochastic* or *deterministic* and finally it can be either *constructive* (i.e. content is generated once) or *generate-and-test* (i.e. content is generated and tested). In addition to the taxonomies provided in [19, 23] — which are applicable to level generators — in this section we put an emphasis on the level design process and derive two more dimensions for clustering level generation approaches. The two dimensions are illustrated under the *user experience model* component of Figure 1.

3.1 First-order vs. Second-order Level Generators

Arguably the level design process as a whole is, by nature, built and driven by emotion. On the one hand there is a *player* that experiences a particular game level. That interaction with the game level elicits affective responses, enables particular cognitive processes and, as a result, yields to a particular playing behavior. Such player emotional responses may, in turn, reflect on the player's bodily reactions (facial expression, posture) or affect changes in the player's physiology. Those affect manifestations caused (in part) by the design of the level can be captured via e.g. physiological sensors or web cameras (see other chapters of this book) and can be used as input to a model that predicts player emotion. Such a model can, in turn be used for personalized level design. In this chapter, we refer to this player-centric approach to emotion-driven level generation *first-order*. On the other hand there is a level *designer* that has particular goals, intentions, preferences, styles and expectations from her design [8]. Most importantly, the level designer incrementally internalizes and builds a high level (or even rather detailed) model of expected player experience during the design process that is used as a design guide. That internal model is tested through piloting, and thorough play-testing. If via testing a mismatch is found between the model of the expected player experience and the actual player experience then two design options are applicable and can even concur: either the level is adjusted accordingly or the designer's expectations and goals about the player experience are altered to match the actual experience. The game emotive goals of the designer and aspects of that internal player experience model can be captured in a similar fashion as with the player. The designer manifests bodily, cognitive and behavioral responses to the design during the design process. Such responses can provide the input to computational representations of the designer's affective, cognitive or behavioral aspects (i.e. *designer models* [8]). We name that

designer-centric approach to player experience design as *second-order* since it is based on an indirect modeling of player experience.

In summary first-order experience-driven level generators build on a *model of player experience*, whereas second-order generators build on a *model of designer experience* which may include intents, goals, styles, preferences and expectations (see Figure 1).

3.2 *Direct vs. Indirect Level Generators*

Further to the distinction between first- and second-order approaches to emotion-driven level generation we also identify two ways in which affect is incorporated in level generation: the *direct* and the *indirect* approach (see Figure 1). According to the *direct* approach the evaluation function of the level generation mechanism is built on a computational model of the player's affect. On the other hand an *indirect* level generator instead considers other aspects of the player experience beyond affect and emotion — such as behavioral traits and cognitive processes. These aspects are seen as proxies of player experience, therefore the *indirect* label. Evidence (as well as common sense) suggests that player (or designer) actions, decisions and real-time preferences are interlinked to experience since the level may affect the player's or the designer's cognitive processing patterns and cognitive focus. As a result, cognitive processes and behavioral patterns may influence emotions and vice versa as cognition, behavior and emotion are heavily interwoven [1]. Thus, one may infer the player's or the designer's emotional state indirectly by analyzing patterns of the interaction and associating user emotions with level context variables [3, 2]. Given the interwoven relation of affect and cognition the boundaries that distinguish between a direct and an indirect level generation approach are often unclear. Figure 1 depicts this relationship via a gradient-colored pattern.

4 Exemplifying Emotion-Driven Level Generation

In this section we outline one critical emotion-driven level generation example per each of the four categories derived from the taxonomy presented in section 3. Within the first-order level generation approaches we describe the *Super Mario Bros* (direct) and the *Mini Dungeons* (indirect) paradigms whereas within the second-order level generators we present the *Sonancia* (direct) and the *Sentient Sketchbook* (indirect) tools for level design.

(a) A *Super Mario Bros* level that maximizes the frustration of a particular player [17].(b) *Mini Dungeons*: A screenshot from a generated level(c) *Sonancia*: A screenshot from a generated level(d) The *Sentient Sketchbook* strategy map design tool

Fig. 2: The example level generators discussed in this chapter.

4.1 *Super Mario Bros*: First-Order, Direct Level Generation

Building on the experience-driven PCG [23] framework, the *Super Mario Bros* level generator employs a *direct* and *first-order* approach to emotion-driven level design. The work of Pedersen et al. [15, 14] and Shaker et al. [18, 17, 16] focuses on the construction of models of player affect via crowdsourced gameplay traces and self-reports of the player experience of several hundred *Super Mario Bros* players. The resulting models fuse behavioral characteristics of gameplay with level parameters and predict aspects of player experience such as player challenge, frustration and engagement. These modes can, in turn, be used to generate personalized emotion-

driven levels by varying the level parameters considered by the player experience models.

More specifically, the work of Shaker et al. [18] — which builds upon and extends that of Pedersen et al. [15, 14] — mines a large set of crowdsourced gameplay data of Super Mario Bros. The data consists of 40 short game levels that differ along six key level design parameters. Collectively, these levels are played 1560 times over the Internet and the perceived experience is annotated by participants via self-reported rankings of engagement, frustration and challenge. The study explores dissimilar types of features, including direct measurements of event and item frequencies, and features constructed through frequent sequence mining. The fusion of the extracted features allowed Shaker et al. to predict reported player experience with accuracies higher than 70%. The models of engagement, frustration and challenge contain level parameters as their input and, thus, are directly applicable for the personalization of game experience via automatic level generation. Exhaustive search within the level parameter space has been used in [16] to achieve that aim.

In addition to the large data corpus of behavioral cues, level parameters and subjective experience annotations a sequel article of Shaker et al. [17] investigated the impact of player *visual cues* (obtained via a webcam) for the construction of player experience models. Obtained results show that when players' visual and behavioral characteristics are fused highly accurate experience models can be constructed as accuracies reach 91%, 92%, and 88% for engagement, frustration, and challenge, respectively. Using exhaustive search on the small level parameter space models can be used to generate a sample of maximally (or minimally) engaging, frustrating, and challenging levels (see Figure 2a).

4.2 *MiniDungeons: First-Order, Indirect Level Generation*

MiniDungeons is a simple turn-based dungeon crawling game, in the style of popular roguelikes such as *Desktop Dungeons* (with similarities to games such as *Rogue* and *NetHack* [4, 5]). The gameplay consists in navigating maze-like dungeons to get from the entrance to the exit of each dungeon (see Figure 2b). Typically, there is more than one way of reaching the end and multiple dead ends. Scattered around the dungeon are monsters, treasures and potions. Monsters sometimes block the path to the exit and need to be overcome to win the level, other times they block paths to treasures or just stand around in the open. Fighting monsters drains health, which can be regained by consuming potions. Importantly, the game can be played in many different ways, depending on whether the player focuses on finishing levels quickly, getting all the treasures, killing all the monsters etc., and also depending on how risk-averse the player is.

Holmgård et al. developed a method for modeling players' behavior in the *MiniDungeons* game (and, by extension, other games featuring tactical decisions) from the perspective of bounded rationality. The model assumes a small number of objectives and takes parameters specifying how important each objective is to the

player. Using neuroevolution, agents can be trained to replicate a player’s style — at least those aspects of player style which are captured by a set of objective weights.

Being able to replicate a player’s playing style is very useful for level generation. Liapis et al. designed a level generator for MiniDungeons based on simulated playthrough [6]. The generator uses evolutionary search for in level space, using playthroughs of levels for evaluating them. By feeding the generator a specific player model, the generator will create levels tailored to the modeled playstyle in the sense that that playstyle is very successful at that level. This is a *first-order* and *indirect* level generator, because while it models the player, it does not model player experience directly; instead, it models the player’s playstyle. It is assumed that the player wants to play in the particular style they exhibit, and therefore that generating levels that make that playstyle successful will increase player enjoyment.

4.3 *Sonancia: Second-Order, Direct Level Generation*

Sonancia [12, 11] is a system built for generating multiple creative domains of horror games, with the intention of creating tense and frightful experiences. *Sonancia* procedurally generates the architecture of a haunted mansion (with rooms and doors which may contain monsters or quest items) as well as the level’s soundscape by allocating audio assets within the rooms and mixing them as the player traverses the level (see Figure 2c). Level generation and soundscape generation are orchestrated by notions of tension and suspense; the level generator attempts to match a designer-specified progression of tension while the sound generator attempts to prompt the player’s suspense in rooms where tension is low.

The *Sonancia* level and soundscape generation system is *direct* as it relies on a function that maps sound and level features to a tension model. The tension-driven level generation is also *second-order* as it explicitly depends on a designer’s provided tension curve — which, in turn, implies the existence of an indirect model of player experience. Further details about the current level and sound generation algorithm behind *Sonancia* can be found in [12, 11].

4.4 *Sentient Sketchbook: Second-Order, Indirect Level Generation*

The last remaining quadrant of our taxonomy is occupied by the *second-order, indirect* level generators. These are generators that model the designer, but not the designer’s affective experience directly. In the following, we will discuss the example of *Sentient Sketchbook* with its designer modeling component.

Sentient Sketchbook is an AI-assisted game design tool for strategy game maps, such as those used in *StarCraft* (Blizzard Entertainment, 1998) [7]. At the core, there is a standard level editor featuring abilities to sketch a strategy map (see Figure 2d). The tool constantly measures the qualities of the current state of the level design

through several metrics related to exploration, area control and balance, and provides real-time feedback to the designer as well as suggestions for changes that the designer can choose to apply or ignore. In the graphical user interface, the various metrics are visualized as meters that give the user instant feedback about e.g. how resource-balanced the current version of the level is, but there is also a visualization in the actual editor pane for e.g. safe resources. The suggestions are presented in a separate panel to the right of the editor, and the user can at any time choose to use a suggestion. These suggestions are generated partly using evolutionary algorithms, starting from the current level and trying to find level variants that satisfy some of the level metrics better.

The designer modeling in Sentient Sketchbook [10] works by constantly tracking the quality metrics of the level as it is being edited. The model then tries to estimate the trend in the editing; essentially, estimate the gradient in multidimensional quality space. This model is then used to influence what suggestions are generated. In a nutshell, the suggestions are generated to mostly lie in the direction the user seems to be pursuing. So if a designer using Sentient Sketchbook seems to be aiming for a more asymmetric map where player A has the resources and player B has the more advantageous terrain, most of the suggestions the tool presents will follow that trend and be even more asymmetric in terms of resources and terrain.

In sum, Sentient Sketchbook with its designer modeling component implements second-order indirect level generation, in that it models the designer's intent and acts on this model. The emotional expression of the levels and the elicited player experience are assumed to be implicit in the intent of the designer, and the model is helping the designer to implement this intent through the generation of suggestions.

5 Discussion

While the examples discussed here come from academic research, it is worth noting that *dynamic difficulty adjustment* is a widespread practice within commercial games of some genres. In particular racing games (such as *Mario Kart 64* (Nintendo, 1996)) frequently adapt their difficulty based on the performance of the player. Some other commercial games include more complex mechanisms; in particular *Left 4 dead* (Valve, 2008) uses a sophisticated dynamic difficulty adjustment mechanism based on tension curves. Player experience, however, is a more complex synthesis of affective and cognitive patterns than mere challenge and only certain aspects of it have been modeled in games. Explicit player emotion-based adaptation exists in commercial games such as the biofeedback-based game *Journey of Wild Divine* (Wild Divine, 2001) for relaxation purposes or the adventure horror biofeedback-enhanced game *Nevermind* (Flying Mollusk, 2015). A number of sensors are available for affective interaction with those games including skin conductance and heart activity. Nevertheless the emotion-based adaptation is realized either through audio-visual aspects or the challenge offered to the player. At the time of writing, we are

not aware of any commercial games that explicitly model player experience for the purpose of generating levels.

In order for emotion-driven game adaptation through level generation to become widespread in commercial-standard games, a number of questions need to be answered, presumably through further research. These questions deal with what features are effective for modeling player experience, how best to generate levels given a particular experience model, and the stability and generality of acquired models. Another critical question is how often particular level attributes should be adjusted. The frequency can vary from simple predetermined or dynamic time windows [21] but adaptation can also be activated every time a new level [16] or a new game [20] starts, or even after a set of critical player actions — such as in *Façade* [13]. The time window of adaptation is heavily dependent on the game under examination and the desires of the game designer.

6 Future Vision and Conclusion

We have outlined a taxonomy of approaches to emotion-driven level generation, elaborating on our existing taxonomy of experience-driven procedural content generation. We have also discussed four examples of emotion-driven level generation, one for each corner in our two-dimensional taxonomy. This is existing work, but it would be safe to say that only very little of the potential of emotion-driven level generation has been realized at this point in time. As so often, further work is needed. Therefore, we would like to conclude this chapter with a brief vision of what a game might look like in the future, when we have figured out emotion-driven level generation sufficiently to make it work reliably on a large-scale commercial-grade game.

You are playing an “open world” game, something like *Grand Theft Auto V* (Rockstar Games, 2013) or *Skryim* (Bethesda Softworks, 2011). Instead of going straight to the next mission objective in the city you are in, you decide to drive (or ride) five hours in some randomly chosen direction. The game makes up the landscape as you go along, and you end up in a new city that no human player has visited before. In this city, you can enter any house (though you might have to pick a few locks), talk to everyone you meet, and involve yourself in a completely new set of intrigues and carry out new missions. If you would have gone in a different direction, you would have reached a different city with different architecture, different people and different missions. Or a huge forest with realistic animals and eremites, or a secret research lab, or whatever the game engine comes up with.

While creating those areas, the game takes your skills, preferences and emotional state into consideration. All of those have been estimated earlier on through recording your interactions with the game, using models of player affect inferred from a large number of players’ interactions in multiple games. So the game might infer that you are bored with the current selection of assassination quests, and venture an educated guess that some opportunities for (in-game) romance might spice things

up. Or decide that you need more, or less, challenge. Or that your aesthetic sense might be stirred by some grand open vistas accompanied by a bombastic score, or maybe a dark claustrophobic basement accompanied by a minimalist electronic tune. Maybe you need more content and activities similar to what you have already experienced; perhaps you had a tough day in the real world and want the comfortable embrace of well-known (yet superficially new) in-game territory and tasks.

Doing all of this right will require enormously wide-ranging and accurate models. How far can we realistically get towards this goal given current technologies and paradigms? We don't know. All we know is that more research is needed. While the methods we have today can already be implemented in constrained domains and controlled environments (e.g., see the Super Mario Bros experiments discussed above) there is no shortage of further research towards making the goal of emotionally adaptive games a reality. In other words, you (and we) have a lot to work on.

Acknowledgements The research was supported, in part, by the FP7 Marie Curie CIG project AutoGameDesign (project no: 630665).

References

1. Antoine Bechara and Antonio R Damasio, *The somatic marker hypothesis: A neural theory of economic decision*, *Games and economic behavior* **52** (2005), no. 2, 336–372.
2. Cristina Conati, *Probabilistic assessment of user's emotions in educational games*, *Applied Artificial Intelligence* **16** (2002), no. 7-8, 555–575.
3. Jonathan Gratch and Stacy Marsella, *Evaluating a computational model of emotion*, *Autonomous Agents and Multi-Agent Systems* **11** (2005), no. 1, 23–43.
4. Christoffer Holmgård, Antonios Liapis, Julian Togelius, and Georgios N. Yannakakis, *Evolving personas for player decision modeling*, *Proceedings of the IEEE Conference on Computational Intelligence and Games (CIG)*, 2014.
5. ———, *Personas versus clones for player decision modeling*, *Proceedings of the International Conference on Entertainment Computing (ICEC)*, 2014.
6. Antonios Liapis, Christoffer Holmgård, Georgios N Yannakakis, and Julian Togelius, *Procedural personas as critics for dungeon generation*, *Applications of Evolutionary Computation*, Springer, 2015, pp. 331–343.
7. Antonios Liapis, Georgios Yannakakis, and Julian Togelius, *Sentient sketchbook: Computer-aided game level authoring*, *Proceedings of the ACM Conference on Foundations of Digital Games*, 2013.
8. Antonios Liapis, Georgios N Yannakakis, and Julian Togelius, *Designer modeling for personalized game content creation tools*, *Proceedings of the AIIDE Workshop on Artificial Intelligence & Game Aesthetics*, 2013.
9. ———, *Computational game creativity*, *Proceedings of the Fifth International Conference on Computational Creativity*, 2014, pp. 285–292.
10. ———, *Designer modeling for sentient sketchbook*, *Computational Intelligence and Games (CIG)*, 2014 IEEE Conference on, IEEE, 2014, pp. 1–8.
11. Phil Lopes, Antonios Liapis, and Georgios N. Yannakakis, *Sonancia: Sonification of procedurally generated game levels*, *Proceedings of the ICCG workshop on Computational Creativity & Games*, 2015.

12. ———, *Targeting horror via level and soundscape generation*, Proceedings of the AAAI Artificial Intelligence for Interactive Digital Entertainment Conference, 2015.
13. Michael Mateas and Andrew Stern, *Procedural authorship: A case-study of the interactive drama façade*, Digital Arts and Culture, 2005.
14. Chris Pedersen, Julian Togelius, and Georgios N Yannakakis, *Modeling player experience in super mario bros*, Computational Intelligence and Games, 2009. CIG 2009. IEEE Symposium on, IEEE, 2009, pp. 132–139.
15. Christopher Pedersen, Julian Togelius, and Georgios N Yannakakis, *Modeling player experience for content creation*, Computational Intelligence and AI in Games, IEEE Transactions on **2** (2010), no. 1, 54–67.
16. N. Shaker, G.N. Yannakakis, and J. Togelius, *Towards automatic personalized content generation for platform games*, Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE). AAAI Press, 2010.
17. Noor Shaker, Stylianos Asteriadis, Georgios N Yannakakis, and Kostas Karpouzis, *Fusing visual and behavioral cues for modeling user experience in games*, Cybernetics, IEEE Transactions on **43** (2013), no. 6, 1519–1531.
18. Noor Shaker, Georgios N Yannakakis, and Julian Togelius, *Crowdsourcing the aesthetics of platform games*, Computational Intelligence and AI in Games, IEEE Transactions on **5** (2013), no. 3, 276–290.
19. Julian Togelius, Georgios N Yannakakis, Kenneth O Stanley, and Cameron Browne, *Search-based procedural content generation: A taxonomy and survey*, Computational Intelligence and AI in Games, IEEE Transactions on **3** (2011), no. 3, 172–186.
20. Georgios N Yannakakis and John Hallam, *Towards optimizing entertainment in computer games*, Applied Artificial Intelligence **21** (2007), no. 10, 933–971.
21. ———, *Real-time game adaptation for optimizing player satisfaction*, Computational Intelligence and AI in Games, IEEE Transactions on **1** (2009), no. 2, 121–133.
22. Georgios N Yannakakis and Ana Paiva, *Emotion in games*, Handbook on Affective Computing (2013), 20.
23. G.N. Yannakakis and J. Togelius, *Experience-driven procedural content generation*, IEEE Transactions on Affective Computing **2** (2011), no. 3, 147–161.