The Evolution of Fun:
Automatic Level Design through Challenge Modeling

Nathan Sorenson and Philippe Pasquier

School of Interactive Arts and Technology,
Simon Fraser University Surrey, 250 -13450 102 Avenue, Surrey, BC
{nds6, pasquier}@sfu.ca

Abstract. A generative system that creates levels for 2D platformer games is presented. The creation process is driven by generic models of challenge-based fun which are derived from existing theories of game design. These models are used as fitness functions in a genetic algorithm to produce new levels that maximize the amount of player fun, and the results are compared with existing levels from the classic video game *Super Mario Bros.* This technique is novel among systems for creating video game content as it does not follow a complex rule-based approach but instead generates output from simple and generic high-level descriptions of player enjoyment.

Key words: video games, challenge-based fun, level creation, genetic algorithms

1 Introduction

Existing processes for creating video game levels are time consuming and expensive, and result in environments that are static and cannot be easily adjusted. Clearly, leveraging a creative system that automatically designs levels would enable independent game developers to generate content that would otherwise require the resources of larger companies. If effective, this process would also result in environments that are not static but are instead amenable to endless variety and modification, ideally creating more interesting and enjoyable experiences for game players. Furthermore, a dynamic and unsupervised method of level creation would allow a virtually limitless amount of content which could be produced off-line and distributed to clients as expansions to the base game, or generated on-the-fly, whereby levels could adapt to individual players as the game is being played.

Although automated methods for creating content are occasionally seen in existing games [1–3], current approaches follow a bottom-up, rule-based approach. This method requires a designer to embed aesthetic goals into a collection of rules, resulting in systems just as difficult to construct as hand-designed levels [4]. Genetic algorithms instead allow developers to specify desirable level properties in a top-down manner, without relying on the specifics of the underlying implementation. However, any effective fitness function for automated level creation must correctly identify the levels that are "fun." To this end, a model of what precisely constitutes a fun level must be developed. In this paper, two such models are proposed: one based on Csikszentmihalyi's concept of "flow" [5] and the other on the notion of "rhythm groups," as described by

Smith et al. [6]. These two models are evaluated by inspecting the levels they produce when employed as fitness functions in a genetic algorithm. These levels are then compared to existing levels that are generally considered to be well-designed. In this case, the automatically generated levels will be compared to four levels selected from the 2D platformer game *Super Mario Bros*.

Generative systems exhibit creativity when they are able to construct solutions that are not simply parameterized variations of existing solutions. Indeed, as Byrne notes, systems that are designed for a specific domain might overlook "solutions that might be applicable to the current problem but are obscured by context- or domain-specific details" [7]. With this in mind, the model of fun that we propose are defined in terms of general theories of player enjoyment instead of by specific details of 2D platformer level design. Although we choose to evaluate the system in the context of this particular genre, our intent is to produce a generative technique that will be applicable to a wide variety of games.

2 Previous Work

2.1 Evolutionary Algorithms in Creative Systems

Evolutionary algorithms are a popular choice for many creative systems [8–10]. The process of creating an initial population of potential artefacts and iteratively evaluating and breeding new populations corresponds closely to the engagement-reflection model of creativity [11]. Furthermore, many evolutionary approaches, such as co-evolution and genetic programming, are not restricted to a well-defined search space but rather re-define the search space as they operate, which is a desirable characteristic for creative systems to have [12].

2.2 Generative Systems in Video Games

Recent attempts have been made to generate novel games without the aid of a human designer. Clark Browne's dissertation [13] considers evolving abstract combinatorial games which are similar to chess and go. His genetic algorithm uses a fitness function consisting of a weighted sum of 57 separate design criteria gleaned from a wide array of sources, including psychology, subjective aesthetics, and personal correspondence with game designers. As the parsimony of the underlying model was clearly not a priority of his research, it is unclear to what degree this approach could be generalized to other contexts.

Togelius and Schmidhuber also use genetic algorithms to generate novel game designs [14]. If a neural net is able to learn to play a particular game well, that design is assigned a high fitness value since it is assumed to contain meaningful patterns that human players would enjoy mastering. In contrast to Browne's approach, this technique is not necessarily bound to a specific type of game. However, this work is currently at a very preliminary stage, and it remains to be seen whether games conducive to machine learning indeed correspond to those enjoyable for human players.

Smith et al. generate levels for 2D platformer games based on a notion of "rhythm groups" [15], which inspires one of the models presented here. Unlike our system,

however, they describe a rule-based system that composes level components together through the use of a generative grammar. This method is crafted specifically for 2D platformer games, whereas our approach seeks greater generality.

2.3 Fun in Video Games

The notion of fun is, without a doubt, incredibly broad, and it is debatable whether the search for a comprehensive definition could ever prove fruitful. Limiting the discussion to the sort of fun characteristic of video games still leaves room for considerable ambiguity. In fact, Hunicke, LeBlanc, and Zubek [16] argue that the term 'fun' in game design must be discarded in favor of more precise terminology. Therefore, they present eight distinct types of pleasure that can arise from game playing: sensation, fantasy, narrative, challenge, fellowship, discovery, expression, and submission. This is not claimed to be a complete categorization, and indeed, many other distinct forms of fun have been identified [17–19].

Such taxonomies can provide useful terminology, but the limited structuring of their categories and the lack of any theoretical underpinning makes it difficult to extract general design principals from them. These categorizations typically identify *what* types of fun can be observed, but they generally do not say *why* such things are fun, and therefore cannot offer principled advice as to *how* designs can evoke a particular type of fun.

2.4 Flow

Many theoretical treatments of fun in games underline the relevance of Csikszentmihalyi's concept of "flow" [5]. Flow refers to a particular state of intense focus, or "optimal experience," that can occur when certain factors are present during a task, such as a sense of being in control, a loss of awareness of self and time, and a close match between the task's difficulty and the individual's skill. This concept has been adapted by Sweetser and Wyath into a game design framework called "GameFlow" [20]. By mapping the factors that encourage the state of flow to the elements of game design, flow can be used as a model for player enjoyment in video games.

Both Koster [21] and Zimmerman and Salen [22] note the relevance of flow to certain game experiences, but neither goes so far as to define fun as equivalent to the state of flow. Certain aspects of the concept are, however, regarded as important for facilitating fun, particularly the requirement that game difficulty should properly correspond to a player's skill. In *A Theory of Fun*, Koster states that "fun is the act of mastering a problem mentally" [21]. It is the process of learning to overcome the challenges inherent in a game that makes the experience enjoyable, whether it be by developing complex strategies to defeat certain opponents in a strategy game or by acquiring the muscle-memory necessary to execute a series of timed button presses in a fighting game. This learning process can be cut short if a game is too easy, as there will be enough problems to be mastered, or if a game is too difficult, as a player will not be able to overcome the problems at all. Similarly, Zimmerman and Salen identify challenge and frustration as "essential to game pleasure" [22].

3 Contribution

Challenge-based pleasure is, therefore, both a category common to all the considered taxonomies and an important aspect of theoretical conceptualizations of video game pleasure. The models explored in this paper will then focus on this particular notion, and can be considered attempts to make explicit the relationship between challenge and fun.

As a simplifying assumption, the models will be developed and verified only within the specific context of the 2D platformer genre, exemplified by *Donkey Kong* [23] and the *Super Mario Bros.* [24] series. As defined by Wolf, these are "games in which the primary objective requires movement through a series of levels, by way of running, climbing, jumping, and other means of locomotion" [25].

This game genre is convenient for many reasons. First, it can be argued that the essential form of pleasure drawn from such games is from being challenged; the core game play mechanic of a platformer game is the dexterity-based challenge of navigating safely from platform to platform. Secondly, the challenges presented in such games assume a uniquely explicit form. Whereas challenge in many games emerges from the dynamic unfolding of an extensive rule set or from competition with an artificially intelligent non-player character [16], the challenge in platform games is embodied in the physical arrangement of the platforms or the positioning of enemies that possess only the most rudimentary artificial intelligence. This embodiment allows the nature of the challenge in the game to be visually inspected, simplifying the analysis. The level design of a platformer game is not merely an environment or container for the game play, but also serves as an essential element of the game play itself. In other words, to understand the nature of challenge in a platformer game, one need look no further than the physical layout of the levels.

3.1 Model Design

Following the hypothesis that challenge is fundamentally related to the pleasure of video games, challenge will therefore serve as the models' primary input variable. As is the case with the notion of fun, challenge is complex and multi-faceted. However, due to the straightforward nature of challenge in 2D platformer games, a reasonable formal conception can be offered. Challenge will be determined entirely by the local configuration of platforms. The challenge of a jump between platforms encountered at time step t, $c(t)$, is typically considered to be proportional to the number of potential player trajectories that successfully traverse the gap [26]. The actual measure used is a rough approximation and is described in (1), where $d(p_1, p_2)$ is the Manhattan distance between the platforms p_1 and p_2 minus the sum of the two "landing footprints," fp, of both platforms (shown in Figure 1) plus a constant.

$$c(t) = d(p_1, p_2) - (fp(p_1) + fp(p_2)) + 2fp_{max} \tag{1}$$

The landing footprint is a measurement of the length of the platform, bounded to the maximum distance a player can jump, fp_{max}. This measure is important, as there

is a much larger margin of error when jumping to a wide platform than a narrow platform, resulting in a less challenging maneuver. The constant $2fp_{max}$ is added simply to ensure that this difficulty measure never produces a negative value.

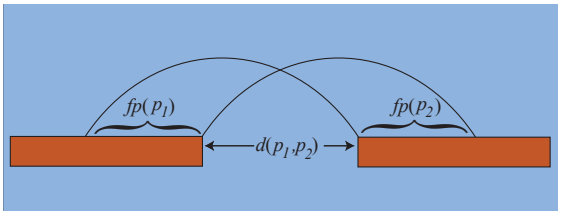

Fig. 1. The landing footprint, fp, is a measure of a jump's margin of error.

To guide the development of the models, four levels from the original *Super Mario Bros.* [24], shown in Figure 2, are taken to be exemplars of the 2D platformer genre. These levels will provide the concrete, empirical data to which the models must conform. Essentially, these levels will constitute an implicit operational definition of the type of fun that is of interest; in an effort to avoid the difficult task of devising an authoritative definition and to remain true to the purpose of mathematical modeling, fun will simply be thought of as a variable that is maximized under the particular configuration of these levels. In other words, any model of challenge-based fun must able to account for the specific patterns of challenge evident in the four selected levels. Ultimately, it is hoped that this sampling will suffice to indicate the validity of the models.

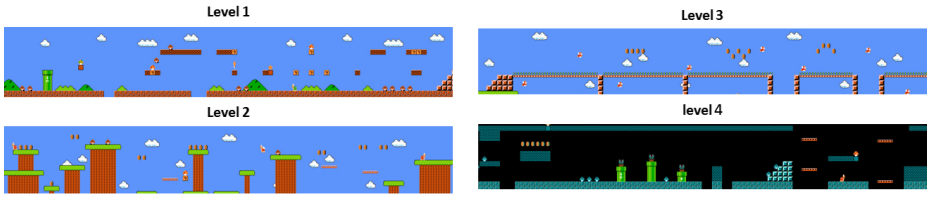

Fig. 2. The four selected *Super Mario Bros.* levels.

3.2 Anxiety Curves

The levels are analyzed with the described challenge metric to produce a characteristic *anxiety curve*. The anxiety curve is the value of the player's anxiety, represented by the variable a, over time. These values are attained by integrating the level's challenge value over the duration of the level, with a constant decay factor, c_{decay} applied, as described in (2). The resulting curve will therefore exhibit an increased slope when there is a sequence of high-difficulty jumps in a short period of time, but will slowly

drop during less challenging segments. The purpose of this function is to highlight the relative dynamics of the challenge over time, as opposed to drawing attention to the actual values of the challenge measurement itself.

$$\frac{da}{dt} = c(t) - c_{decay} \tag{2}$$

The placement of enemies in the *Mario* levels is certainly important to the challenge dynamics. Though our difficulty metric does not explicitly refer to enemies, we are able to capture this information by internally representing each enemy as a small gap in the level. Essentially, the enemies are considered to be as difficult as a small jump between platforms. Since enemies must generally be avoided or defeated through jumping, this technique serves as a rough approximation for the challenge a player faces when encountering an enemy.

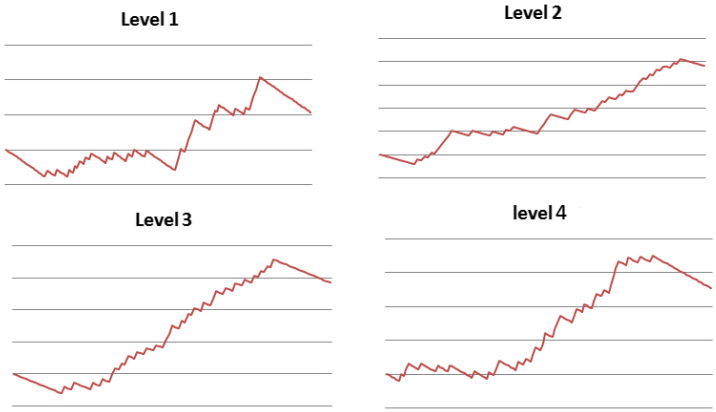

Fig. 3. *Super Mario Bros.* anxiety curves, charting anxiety, a, (vertical axis), over time, t, (horizontal axis).

As Figure 3 demonstrates, there are recognizable similarities between the anxiety curves of the four *Super Mario Bros.* levels. Although a crude challenge measurement is used and the conceptualization of anxiety is primitive, some structure can already be qualitatively identified: each level begins and ends with a phase of decreasing anxiety and has a recognizable period of high anxiety directly preceding the end. Three of the levels exhibit a lower anxiety slope during the beginning half of the level, followed by a higher slope. Finally, all four levels exhibit a sawtooth pattern.

3.3 Flow-based Model

The first model is a naïve interpretation of Csikszentmihalyi's often-cited concept of flow, specifically the portion most often applied to game design: the necessary match

between game challenge and player skill. Intuitively, then, the amount of fun had by a player must decrease as the mismatch between skill and challenge increases. This relationship is expressed in (3).

$$\frac{df}{dt} = -(c_{skill} - c(t))^2 \tag{3}$$

Fun is represented by the variable f, and like anxiety, it accumulates over time and is therefore considered differentiable. Skill (c_{skill}) is assumed to remain constant over the duration of a single level, and can be interpreted as the greatest level of challenge a player is able to effectively overcome. It is clear that to maximize the value of f, c_{skill} and $c(t)$ must be kept as close as possible, which is precisely what the state of flow requires. Therefore, because we are concerned only with maximization of fun, the fact that this model produces negative values is of no consequence.

3.4 Genetic Algorithm

To evaluate this model, a genetic algorithm is used to automatically create a level that maximizes the amount of fun (as defined by the model) experienced during play. Each individual level is encoded as a genotype consisting of a sequence of x, y coordinates. Each coordinate pair specifies the horizontal and vertical pixel offset of a 64 by 64 pixel platform relative to the preceding platform. The advantage of relative position encoding is that the range of these offsets can be limited to the maximum distance a game character can jump. As no playable level will consist of a platform that is unreachable from the previous platform, all valid levels can be represented in this manner (assuming, of course, that levels are linear with no branching paths). As well, all unplayable levels are eliminated from the evolutionary search space with this encoding, greatly improving the performance of the genetic algorithm.

Mutation consists of perturbing a coordinate by a normally-distributed random value, and individuals are combined through variable-point crossover to allow for various sizes of genotypes. Since two levels of the same total length could consist of different numbers of platforms, therefore requiring different numbers of x, y pairs in the genotype, a variable-length genotype is necessary. The fitness function is a straightforward implementation of (3), which aims to maximize the amount of fun accumulated. As well, levels are generated to be a specific total length and are heavily penalized for deviating from this externally imposed length. This restriction is necessary to avoid the evolutionary search from simply evolving longer and longer levels as a means of accumulating arbitrarily high amounts of fun. The population consists of 200 individuals that are evolved for 10,000 generations. Although efficiency is not presently a concern, running time was on the order of several hours on a mid-range dual-core PC.

A generated level and its corresponding anxiety curve are shown in Figure 4. This level appears to be a chaotic scattering of platforms, and the anxiety curve seems to be of a nearly-constant slope. While it does not correspond to the *Super Mario Bros.* levels, this shape is to be expected; in accordance with the concept of flow, challenge is kept as closely as possible to a constant value in levels with an anxiety curve of constant slope. Therefore, in its naïve application, flow likely does not serve as an effective model for

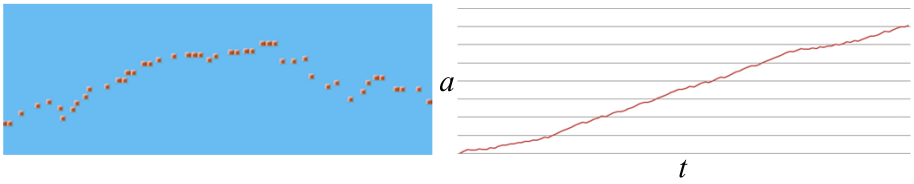

Fig. 4. Level and anxiety curve for flow-based model.

challenge in video games. This result agrees with the findings of those who believe flow is an inappropriate guideline for game design, such as Juul [27] and Falstein [28].

3.5 Periodic Model

Clearly, variation in a level's challenge is desired. Indeed, this property is inherent in the notion of "rhythm groups," as described by Smith et al. [6] in their structural analysis of the architecture of platform games. A rhythm group is considered a fundamental unit in platform level design, and consists of a moment of high challenge followed by a moment of low challenge. Rhythm groups can explain the oscillatory anxiety curves seen in the *Super Mario Bros.* levels introduced in Section 3.1, and in the attempt to encourage such rhythmic variation, a new model is described in (4).

$$\frac{df}{dt} = m * \frac{da}{dt} \tag{4}$$

The behavior of this model is determined by the m variable, which can take the value $+1$ or -1. When positive, the player will accumulate fun at the same rate that the anxiety increases. This state represents the pleasure gained from being challenged. However, when the anxiety becomes too great, that is, when $a > a_{upper}$ where a_{upper} is some constant threshold, m will become negative. After this point, fun can only be accrued as anxiety falls. When the level of anxiety becomes low enough, that is, $a < a_{lower}$, m will again become positive and the player will be ready for new challenges. Whereas c_{skill} represents, in the flow-based model, a particular degree of challenge, a_{upper} and a_{lower} here specify levels of anxiety (essentially, challenge integrated over time, as described in (2)). This model is likewise used as a fitness function in an evolutionary run, and Figure 5 depicts the resulting level and its anxiety curve.

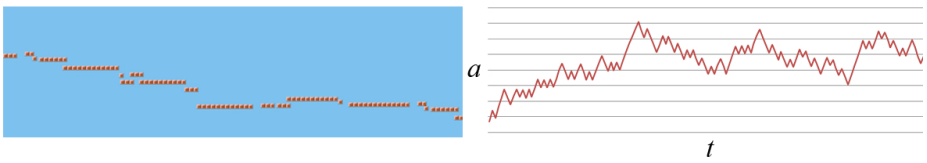

Fig. 5. Level and anxiety curve for periodic model.

This level qualitatively appears much more desirable than the previous one, with clearly identifiable rhythm groups. The anxiety curve likewise exhibits an oscillatory nature. However, the observed pattern does not possess a unique high-anxiety peak near the end of the level; rather its cyclical appearance is much more regular than what is observed in the *Super Mario Bros.* levels.

4 Discussion and Future Work

The approach taken in the development and evaluation of this system prompts several observations. First, by employing theoretical accounts of fun in video games, we are able to identify abstract patterns in existing levels: rhythm groups, as described by Smith et al., can be identified in the anxiety curves, and a characteristic dramatic arc can be seen in all the levels. Secondly, we generate levels in a top-down manner by using high-level models as fitness functions in a genetic algorithm. This technique is a promising alternative to the time-consuming trial-and-error approach associated with the creation of rule-based systems. Finally, the analyses of existing levels and the corresponding models were conducted with regard to the dynamics of challenge over time, not in terms of specific details of 2D platformers. This generality allows for the possibility of extending this generative technique to other games and other genres.

With these promising initial results, we intend to further explore the utility of this top-down approach for level design. A clear first step would be to extend the genetic algorithm to generate enemies and moving platforms. It would then be interesting to develop challenge metrics for other games and compare the resulting anxiety curves, looking for similarities and differences between game genres. As well, it might prove useful to augment the genetic algorithm with meta-evolutionary techniques, such as evolving the encoding of the levels or the fitness functions. These techniques could further reduce the influence of the human designer in the construction of the content, relegating even more creative responsibility to the underlying system.

Although this system does not yet create output comparable to the work a professional level designer, it can be considered an exploratory first step toward that goal. Much work still needs to be done regarding the formal analysis of existing games and the specification of exactly what variables are important when predicting player enjoyment. Even if a definitive formalism is unlikely, it is hoped that the very process of identifying simple and general models of fun will enable creative systems to enhance the practice of game design.

References

1. The NetHack DevTeam: Nethack. (2009) `http://www.nethack.org/`.
2. Meier, S.: Civilization. MicroProse (1991)
3. Matsuura, M.: Vib-Ribbon. Sony Computer Entertainment (1999)
4. Remo, C.: MIGS: Far Cry 2's Guay on the importance of procedural content. Gamasutra (2008) `http://www.gamasutra.com/php-bin/news_index.php?story=21165`.
5. Csikszentmihalyi, M.: Flow: The Psychology of Optimal Experience. Harper Perennial (1991)

6. Smith, G., Cha, M., Whitehead, J.: A framework for analysis of 2d platformer levels. In: Sandbox '08: Proceedings of the 2008 ACM SIGGRAPH symposium on Video games, New York, NY, USA, ACM (2008) 75–80
7. Byrne, W., Schnier, T., Hendley, R.J.: Computational intelligence and case-based creativity in design. In: Proceedings of the International Joint Workshop on Computational Creativity 2008, Madrid, Spain (2008) 31–40
8. Sims, K.: Artificial evolution for computer graphics. In: SIGGRAPH '91: Proceedings of the 18th annual conference on Computer graphics and interactive techniques. Volume 25., ACM Press (1991) 319–328
9. Dipaola, S., Gabora, L.: Incorporating characteristics of human creativity into an evolutionary art algorithm. Genetic Programming and Evolvable Machines **10**(2) (2009) 97–110
10. Kicinger, R., Arciszewski, T., Jong, K.D.: Evolutionary computation and structural design: A survey of the state-of-the-art. Computers & Structures **83**(23-24) (2005) 1943–1978
11. Jorge, P.M., Tavares, J., Cardoso, A., Pereira, F.B., Costa, E.: Evolving creativity. In: Computational Creativity Workshop, 7th European Conference in Case Based Reasoning, Berlin, Germany, Springer (2004) 91–102
12. Boden, M.: The Creative Mind; Myths and Mechanisms. Routledge (2003)
13. Browne, C.: Automatic generation and evaluation of recombination games. PhD in computer science, Queensland University of Technology, Brisbane, Australia (2008)
14. Togelius, J., Schmidhuber, J.: An experiment in automatic game design. In: IEEE Symposium on Computational Intelligence and Games. (2008) 111–118
15. Smith, G., Treanor, M., Whitehead, J., Mateas, M.: Rhythm-based level generation for 2d platformers. In: FDG '09: Proceedings of the 4th International Conference on Foundations of Digital Games, New York, NY, USA, ACM (2009) 175–182
16. Hunicke, R., LeBlanc, M., Zubek, R.: MDA: A formal approach to game design and game research. In: Challenges in Game AI Workshop, Nineteenth National Conference on Artificial Intelligence. (2004) 1–5
17. Apter, M.J.: A structural-phenomenology of play. In Kerr, J.H., Apter, M.J., eds.: Adult Play: A Reversal Theory Approach. Swets and Zeitlinger, Amsterdam (1991) 18–20
18. Malone, T.W.: What makes things fun to learn? heuristics for designing instructional computer games. In: SIGSMALL '80: Proceedings of the 3rd ACM SIGSMALL symposium and the first SIGPC symposium on Small systems, New York, NY, USA, ACM Press (1980) 162–169
19. Garneau, P.A.: Fourteen forms of fun. Gamasutra. October 12 (2001)
20. Sweetser, P., Wyeth, P.: Gameflow: a model for evaluating player enjoyment in games. Comput. Entertain. **3**(3) (2005) 3
21. Koster, R.: Theory of Fun for Game Design. Paraglyph Press (2004)
22. Salen, K., Zimmerman, E.: Rules of Play : Game Design Fundamentals. The MIT Press (2003)
23. Miyamoto, S.: Donkey Kong. Nintendo (1981)
24. Miyamoto, S., Yamauchi, H., Tezuka, T.: Super Mario Bros. Nintendo (1987)
25. Wolf, M.J.P.: The Medium of the Video Game. University of Texas Press (2002)
26. Compton, K., Mateas, M.: Procedural level design for platform games. In: 2nd Artificial Intelligence and Interactive Digital Entertainment Conference. (2006) 109–111
27. Juul, J.: Fear of failing? the many meanings of difficulty in video games. In Yao, X., Burke, E., Lozano, J.A., Smith, J., Merelo-Guervs, J.J., Bullinaria, J.A., Rowe, J., Tino, P., Kabn, A., Schwefel, H.P., eds.: The Video Game Theory Reader 2. Routledge, New York (2009) 237–252
28. Falstein, N.: Understanding fun–the theory of natural funativity. In Rabin, S., ed.: Introduction to Game Development. Charles River Media, Boston (2005) 71–98