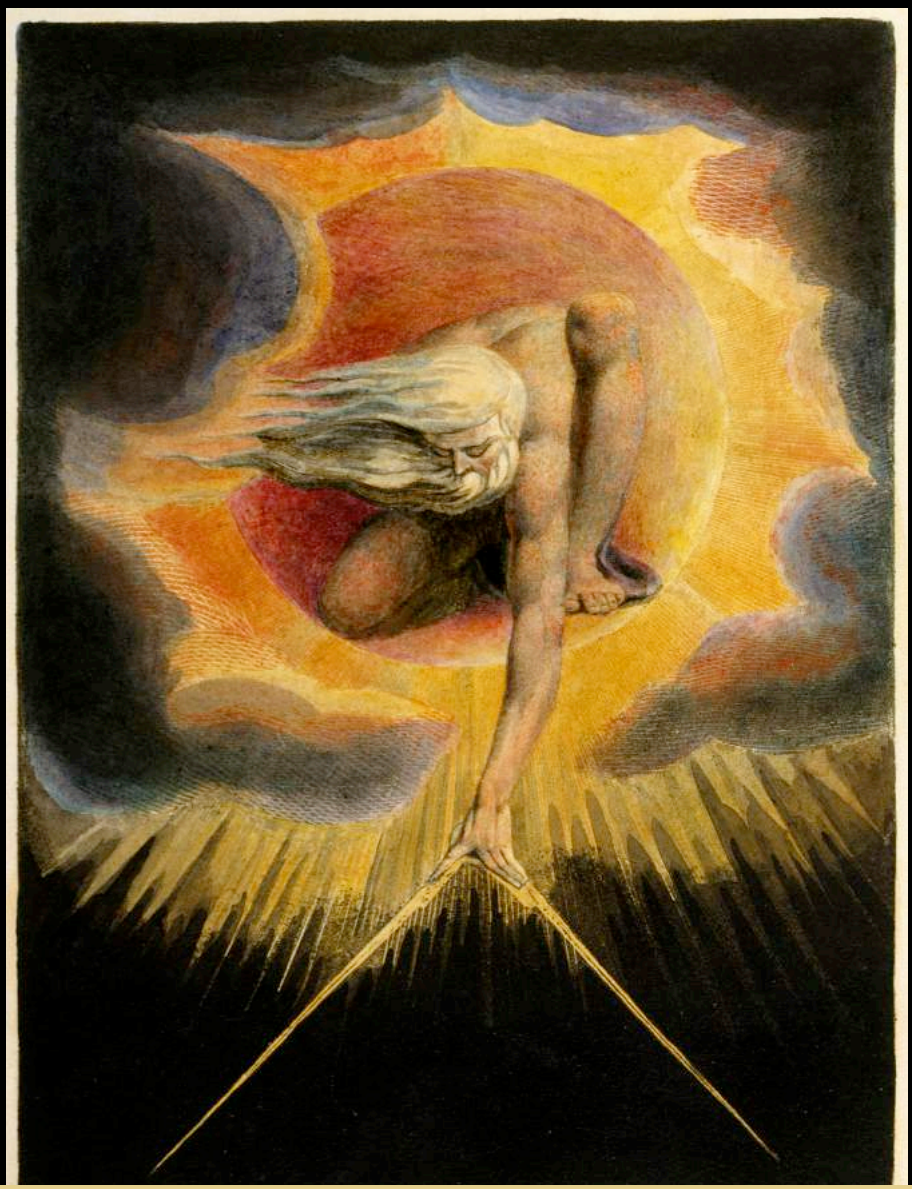

Rui Alexandre Neves Craveirinha

The Authorial Game Evolution Tool for Player Experience Design

Tese de Doutoramento do Programa de Doutoramento em Ciências e Tecnologias da Informação,
orientada pelo Professor Doutor Licínio Gomes Roque e apresentada
à Faculdade de Ciências e Tecnologia da Universidade de Coimbra

Julho 2017

· U C ·

UNIVERSIDADE DE COIMBRA

UNIVERSIDADE DE COIMBRA

DOCTORAL THESIS

The Authorial Game Evolution Tool for Player Experience Design

RUI CRAVEIRINHA

Submitted in fulfilment of the requirements
for the degree of PhD

Centro de Informática e Sistemas da Universidade de Coimbra
Departamento de Engenharia Informática

Supervised by
DR LICÍNIO ROQUE

February 27, 2018

"Before defining art or any concept we must answer a far broader question. What's the meaning of man's life on earth? Maybe we are here to enhance ourselves spiritually. If our life tends to this spiritual enrichment, then art is a means to get there. This is in accordance with my definition of life. Art should help man in this process. Some say that art helps man to know the world like any other intellectual activity. I don't believe in this possibility of knowing; I am almost an agnostic. Knowledge distracts us from our main purpose in life. The more we know the less we know; getting deeper, our horizon becomes narrower. Art enriches man's own spiritual capabilities and he can then rise above himself to use what we call "free will"."

Andrei Tarkovsky, answering "What is Art?"

Cover Image: Blake, William. *Europe: a Prophecy*. 1794. British Museum, London

UNIVERSIDADE DE COIMBRA

Abstract

Faculdade de Ciências e Tecnologia da Universidade de Coimbra
Departamento de Engenharia Informática

PhD

The Authorial Game Evolution Tool

by RUI CRAVEIRINHA

Video game design and production best practices frequently involve an iterative process of prototyping and evaluation through play-testing, with the goal of fine-tuning player experience. Such a process can be difficult, long, costly, and there is not a solid theoretical background to support it. To aid matters, we propose the Authorial Game Evolution tool (or AGE), a system that lets designers use procedural content generation to explore and optimize video games so they mediate an intended form of player-experience.

AGE's working is simple: designers take a base-game and define a number of Game Variations that a procedural content generator can operate on. At the same time, they define Design Goals, by establishing target values for a number of experience indicators. AGE then has play testers play procedurally generated variations of that game, until it finds one which successfully mediates the target experience. This empowers designers by affording them a more effective way to craft video games that mediate the player experiences they want.

This dissertation focuses on AGE and the Design Science Research process used to design it. The critical readings of scientific literature that informed it, as well as the experiments in which its design was studied and refined are herein detailed.

The first experiment consisted of using AGE to solve a hypothetical design problem: finding a good combination of values for 8 game design parameters so as to achieve 3 player experience indicator values. Employing a simple search algorithm for generation of new parameter values, and using 25 players to test the end results, AGE was able to generate games whose experience indicators closely matched the intended target values. Thus, results indicate this approach can be used to solve simple game design problems.

Two Participatory Design sessions followed where game designers developed a prototype of the AGE interface. Discourse analysis from the sessions shows that designers struggled to envision how they could design games with AGE, and had difficulties in understanding its terminology and mode of operation. Additionally, designers proposed several changes to AGE, key among this, the proposal of an Exploration use-case, where instead of seeking to optimize player experience, AGE would be employed to try out different variations of the same game, mapping out the surrounding design space. Identification of these issues was used to further revise the tool, its mode of use, and interface, leading to the development of a complete prototype.

Finally, a case-study was held that illustrates how creative game design supported by AGE might unfold. A designer took a base-game and used AGE to evolve it into a novel game throughout four design sessions. However, he appropriated it exclusively for exploration of the base-game (and not optimization); despite this, he showed interest in the tool for optimization of games. Creativity Support Index self-report shows the designer found AGE 'Very Good' in supporting his design, particularly for exploration of the design-space. Besides this, the designer also found the tool useful for structuring the design process, warranting this change to his design practice as valuable in of itself.

Together, results from these experiments establish the tool's feasibility and potential usefulness. Additionally, we report several findings with respect to how procedural content generation methods might be framed in game design processes – including a taxonomy that clarifies human and computational actor roles, and the recurring theme of Exploration as a use-case beyond player experience optimization – as additional scientific contributions from this project.

Keywords: User Experience, Video Games, Game Design, Procedural Content Generation

UNIVERSIDADE DE COIMBRA

Resumo

Faculdade de Ciências e Tecnologia da Universidade de Coimbra
Departamento de Engenharia Informática

PhD

The Authorial Game Evolution Tool

por RUI CRAVEIRINHA

As melhores práticas de design e produção de videojogos frequentemente envolvem um processo iterativo de prototipagem e avaliação através de playtesting, cujo objectivo é afinar a experiência de jogador. Este processo pode ser difícil, longo, caro, e não existe uma base teórica sólida para o suportar. Para amenizar o processo, propomos uma ferramenta de Evolução Autoral de Jogos (de nome AGE), um sistema que permite a designers usar geração de conteúdo procedimental para explorar e optimizar videojogos, de forma a que estes medeiem uma forma pretendida de experiência de jogador.

O funcionamento do AGE é simples: designers pegam num jogo base e definem um número de Variações de Jogo sobre as quais um gerador de conteúdo procedural pode operar. Simultaneamente, definem Objectivos de Design ao estabelecer valores alvo para um número de indicadores de experiência. O AGE deixa então que jogadores experimentem variações do jogo original geradas procedimentalmente, até encontrar uma que medeie com sucesso a experiência alvo. Tal capacita os designers ao fornecer-lhes uma forma mais eficaz de criar video jogos que medeiam as experiências de jogador que eles pretendem.

A presente dissertação foca-se no AGE e no processo de Investigação Científica por Design realizado para o desenhar. São aqui detalhadas as leituras críticas da literatura científica que o informaram, bem como as experiências em que o mesmo design foi estudado e refinado.

A primeira experiência consistiu em usar a abordagem AGE para resolver um problema de design hipotético: encontrar uma boa combinação de valores para 8 parâmetros de design, de forma a atingir 3 valores para indicadores de experiência. Empregando um algoritmo de busca simples para a geração de parâmetros, e usando 25 jogadores para testar resultados, o AGE foi capaz de gerar jogos cujos indicadores de experiência se assemelhavam com os valores alvo pretendidos. Assim, os resultados indicam que esta abordagem pode ser usada para resolver problemas de design simples.

Duas sessões de Design Participatório seguiram-se onde designers de jogo desenvolveram um protótipo do interface do AGE. Análise do discurso destas sessões demonstra que os designers debateram-se com a necessidade de imaginar em como usar o AGE para desenhar jogos, e tiveram dificuldades em entender a sua terminologia e forma de funcionamento. Adicionalmente, os designers propuseram diversas alterações ao AGE; fundamentalmente, a proposta para um caso de uso de Exploração, onde ao invés de optimizar a experiência do jogador, AGE seria usado para experimentar diferentes variações do mesmo jogo, mapeando assim o espaço de design circundante. A identificação destes problemas foi usada para rever a ferramenta, o seu modo de uso, interface, que culminou no desenvolvimento de um protótipo completo.

Finalmente, um caso de estudo foi realizado que ilustra como processo criativo de design de jogo suportado pelo AGE se pode desenvolver. Um designer pegou num jogo-base e usou AGE para o evoluir para um novo design de jogo ao longo de 4 sessões de design. No entanto, ele apenas o apropriou para a exploração do jogo-base (ao invés de optimização); apesar desse facto, demonstrou interesse na ferramenta para a optimização de jogos. Resultados de um relatório de auto-avaliação (denominado Index de Suporte à Criatividade) indicam que achou o AGE 'Muito Bom' no suporte à sua actividade criativa, particularmente no que concerne a exploração do espaço de design. Mormente, o designer achou a ferramenta útil para a estruturação do processo de design, considerando tal mudança à sua prática de design como valiosa em si.

Juntos, estes resultados estabelecem a aplicabilidade da ferramenta, bem como a sua potencial utilidade. Reportamos ainda diversas contribuições adicionais sobre como métodos de geração procedimental podem ser enquadrados em processos de design de jogo – incluindo uma taxonomia que clarifica os papéis dos actores humano e computacional, e o tema recorrente da Exploração enquanto caso de uso além da optimização de experiência de jogador.

Palavras-Chave: Experiência do Utilizador, Videojogos, Design de Jogo, Geração de Conteúdo Procedimental

Acknowledgements

To my girlfriend, partner and greatest friend, Claudia Madeira, and to my family, especially my Parents, for their immense love, support, and for always being there,

To Supervisor Licinio Roque, for guiding me in this arduous journey.

To my PhD colleagues and friends – especially Luís Pereira, Filipe Penicheiro, Nuno Barreto, and Valter Alves – for their companionship, work and insight,

To all my padawan slaves... pardon, my "undergrad students", especially João Soares and Tiago Agostinho, for all their help and hard work in helping make this work a reality,

To my friends, Alexandre Martins, João Tojo, Jorge Sousa, Luís Pureza and Miguel Silva, for their companionship, ideas, feedback, and dutiful role as testers of all kinds of broken prototypes,

To my other sensei, Bruno de Figueiredo, for opening my eyes to all things that lie in that unexplored gulf between video games and art,

To all these who helped me on this journey, and to all others who I may have forgot to mention, I thank from the bottom of my heart for their support.

This thesis and all related activities were partially financed by PhD Scholarship number SFRH/BD/75196/2010, awarded by the Fundação de Ciências e Tecnologia.

Contents

List of Figures

Chapter 1

Introduction

Despite their success, video games have long been struggling with their coming of age. Thought of as a potential art form at least since 1982 by game designers such as Chris Crawford (1982), their legitimacy as such has been slow in reaching acceptance. Until very recently, the lack of creativity and maturity present in most titles was considered worrisome (Salen and Zimmerman, 2004) and despite their commercial and cultural relevance (Poole, 2000; Frasca, 2001), they were often perceived as immature, a past-time for the young, devoid of greater purpose (Juul, 1999; Bogost, 2007; Poole, 2000).

In historical terms, video games are a very recent phenomenon. The very first artefact to be consensually considered a video game, William Higinbotham's "Tennis for Two" is from 1958 (Herman, 2001). The game design discipline is younger still (Salen and Zimmerman, 2004), and scientific research only officially started in 2001 (Arseth, 2001). Both the design and scientific fields of video game study are new. Designing video games that mediate a given experiential effect to players is still a daunting and challenging task to accomplish, and, as we will argue in the next chapter, there is not enough stable theory to guide this process. And yet, such hard to achieve goal is the watermark that good game designers should aspire to (Fullerton, Swain, and Hoffman, 2008).

Part of the difficulty in mediating player experience arises from video games inherent interactivity, which makes the way they are experienced by players vastly unpredictable (Hunicke, Leblanc, and Zubek, 2004). Being interactive means different players can potentially see different parts of the same video game, in a different order, in a different context, therefore having potentially different experiences. Consequently, best design and development practices involve a long and costly process of iterative cycles of development and testing (Fullerton, Swain, and Hoffman, 2008), intent on assuring the desired experience

is mediated. The research project covered by this thesis aims to help solve this problem, by designing, developing and researching a tool that can empower designers to more effectively mediate their intended forms of player experience.

Several advanced experience evaluation methods are already used in the process (Mahlmann et al., 2010), affording designers vast amounts of data which can then be used to test out whether player experience is in accord with their goals or not, and consequently what aspects of the design need revision to steer experience in the intended direction. We hypothesize that further improvements to this process could be made if experience data collection could be framed in tools that, based on such data, could automatically evolve games' design so they end up mediating the intended player-experience. Such a move should, in theory, serve to simultaneously alleviate the development process – by automating part of it – while increasing the likelihood of designers producing games that mediate their intended player-experience.

Close to the aforementioned intents, we found a family of algorithmic approaches entitled Experience-Driven Procedural Content Generation. Procedural Content Generation (PCG) is the creation of game content by automatic or semi-automatic algorithms (Hendrikx et al., 2013) and its Experience-Driven variants (EDPCG) generate content meant to optimize player-experience (Yannakakis and Togelius, 2011). These methods could help in improving the game design process; their success in optimizing content for certain emotional patterns in players showing great potential (Shaker, Yannakakis, and Togelius, 2010; Pedersen, Togelius, and Yannakakis, 2010). However, reviewed EPDCG studies did not sufficiently focus on a key issue to this problem: how can designers interface with these methods, in ways that preserve their authorial intent?

1.1 The Authorial Game Evolution Tool

The issue with using EDPCG techniques (if one frames them with the goal of improving game design processes) is that because they automate creative tasks, they tend to remove designers' agency. This creates a dual problem. On one side, it is unlikely that creators will look kindly on such techniques if they lose control over their creative output, and this in turn will make them less prone to adopt them. On the other side, giving part of the creative burden to unsupervised algorithms can potentially decrease the quality and uniqueness of the end-result, as PCG is infamously known for delivering comparatively dull and repetitive content (Tutenel et al., 2009). Thus, in order to use these techniques to solve our problem, it is our view that there needs to be a way to frame these procedural techniques in a way that balances designers authorial agency over the end-result, with the utility of letting a machine take part of the work's burden.

The first main research question that guided us was *How can game designers use experience evaluation methods and procedural content generation algorithms to prototype video games that mediate intended forms of player experience?* Our answer to this is AGE, or The Authorial Game Evolution Tool, a tool designed precisely to solve this question. AGE seeks to utilize automatic generation and evaluation of artefacts to increase both the efficiency and effectiveness of the design process, while empowering authorial exploration of the design space, and furthering designers' creative agendas for their artefacts.

In order to improve game design and development processes, our idea was to harness the increased automation PCG can offer in service of an *ad-hoc* human agenda, hoping that in this way human and machine could complement each other and arrive at an interesting equilibrium. So instead of having a method that could autonomously optimize a given experiential trait for a given type of content, we thought what was needed was a tool that allowed human designers to select what experience they want to optimize and then have PCG systems do so.

The greatest challenge here was in finding a way that non-technical designers could communicate their design agendas for player-experience to a computational system, and then have *it* find the content to fulfil said agenda. To preserve designers authorial agency, the approach should strive to be as customizable and content-agnostic as possible, allowing game creators to optimize (potentially) any type of player-experience for any game content.

The solution we have researched is called The Authorial Game Evolution (or AGE) tool [1]. To use it designers must start with a base-game they wish to evolve. After integrating it with the AGE tool, they define a target goal for player-experience, and then let a search algorithm find the best variation of their base-game that mediates said experience to play testers. Content is generated based on rules defined by designers called 'Game Variations', which define how core game parameters are varied in procedural generation. A hypothetical example of a Game Variation might be, *generate levels with obstacle numbers between 5 and 10*, to which the generator must then create different levels with 5, 6, 7, 8, 9 and 10 obstacles.

Play-testers' play generated content and the resulting experience is evaluated to find out if it is in accordance to designers wishes. Experience evaluation is based on a number of designer defined tests named 'Design Goals'. A Design Goal is a test with a simple IF-THEN-ELSE logic applied to an experience indicator; if the indicator passes the test, content gets a positive score, otherwise it either gets a null or negative score. For instance, one might want to test how many times the player dies in the obstacle-ridden levels, by having a test like *IF number of player deaths in the level is below 2, THEN score = 100, ELSE score = 0*. These rules are converted into a fitness function which the search algorithm will use to optimize content.

By providing a way designers can, with any authorial intent, control both how the procedural generation works and the experiential end-result, their agency is preserved. Because part of the design and production process is automated, then there is the potential for an improvement of the process, whilst at the same time, offering greater guarantees that the desired experience is mediated to sample of play-testers (as long as the search-algorithm is capable of finding a good result).

If one can verify that AGE is a valid answer to the first research answer – that it can be used in game design processes as a way to achieve player experience mediation, then this naturally entails another equally relevant question: *How does the introduction of AGE transform game design processes?* The answer to this question assesses the impact the tool might have in the game design process.

[1] Throughout this thesis, AGE's working and the names of its components have changed, what follows is a description referring to the final version of the tool.

1.2 Research Summary

To establish if the AGE approach to experience optimization was feasible, we started with a proof of concept. We took a hypothetical game design problem – create a speed-run like Mario – and converted it into a search-space problem: finding the right set of values for 8 game design parameters so as to achieve 3 player experience indicator values. Employing a Genetic algorithm for generation of parameter values, and using 25 players to test the end results, AGE was able to generate content whose experience indicators closely matched the intended target values. For two of the three indicators, one could take the play-tester data and generalize the indicators averages as indicative of means sufficiently close to intended values. This served as a proof of concept on the possibility of design problems being translated into AGE's framework, and at the same time, it showed the approach could be used for finding solutions to simple design problems even when paired with out of the box search algorithms.

Given AGE could be used to solve design problems, it warranted design of a full prototype so as to test it in action. As our goal was to provide game designers with a tool that they would find useful for their creative tasks, it made sense to let designers themselves dictate how the design should be. As such, we held two Participatory Design sessions in which a group of designers crafted an interface prototype for the AGE approach. Two results came out of these sessions: one was a design for the AGE interface, and the other was an analysis of designers speech and work while designing the interface.

Discourse analysis from the sessions shows that designers struggled to envision how they could design games with AGE, and had difficulties in understanding both its terminology and mode of operation. Additionally, designers proposed several changes to AGE itself, key among this, the proposal of an Exploration use-case. In this case, instead of seeking to optimize player experience, AGE would be employed without Design Goals, merely trying out different variations of the same game, and mapping out the design space surrounding the base-game. Identification of these issues was used to further revise the tool. The new use case was integrated, showing a possible answer to the question of how AGE would impact design processes, well beyond our original intent of player experience improvement. The core terminology was revised under a new metaphor to ease in comprehension of the AGE approach; this was validated and further refined in a usability study. This reinforces the fact that the greatest challenge in using the approach lies in finding the right interface for designers.

Finally, as a full prototype of the AGE tool was complete, it needed to be tested in action. A case-study was then realized to illustrate how creative game design supported by AGE might unfold. A game designer took a previously existing base-game and used AGE to evolve it into a novel gamexw. Throughout four design sessions he used AGE to tinker his game, manipulating it directly and altering it procedurally by changing Game Variations and Design Goals. Surprisingly given our initial expectations, he appropriated it exclusively for Exploration of the base-game (and never for optimization). Despite this, the designer showed verbal interest in the tool for optimization of games, just not for the game design he ended up realizing in the case-study.

The designers' Creativity Support Index self-report showed he found AGE 'Very Good' in supporting his design, assuring us that AGE was capable of assisting his design process, and therefore that it is a viable answer on how to appropriate procedural content generation methods for game design processes. In accordance with his work, highest scores of the self-report concerned the exploration component of the process; once again, this reaffirmed the pertinence of the Exploration use case afforded by procedural content generation methods, and suggested that AGE's (initial) focus on optimization was not adequate for the designer community.

As an additional outcome of the case-study, the designer also expressed that he found the tool to be useful for structuring the design process. By forcing designers to define problems in this novel way, asking them for a formal definition of Design Goals for player experience, it structured his design activity into cycles of problem setting and problem solving. He warranted that this change to his design practice as valuable in of itself, thus serving as a secondary benefit to the use of the AGE tool.

Taken as a whole, these studies show that AGE is a viable contribution towards improving contemporary game design processes. It shows how experience evaluation methods and procedural content generation algorithms can be appropriated in a tool and process that improve existing design practices, all while empowering designers authorial agency over the end-result. The remainder of this thesis will elaborate on these points and further detail both the AGE tool, the experiments realized to design and evaluate it, and the nature of the diverse scientific contributions that result from this research project.

1.3 Document Structure

The remaining chapters in this thesis are organized in the following manner. Chapter 2 further details our critical reading of existing scientific literature, justifying our goal, motivation, and framing many of our scientific and design decisions. It starts by explaining the relevance of player-experience in game design, laying the ground work for establishing the benefit that player-experience optimization can have in video game design and production. It lists major problems faced by the game design community, as well as the value and limitations that PCG techniques have had in tackling these. Several PCG studies are discussed, providing the background that informed our own take on the problem.

As a side result from the literature review and our positioning in this field, a contribution was made on how to classify Procedural Content Generation approaches in terms of their Human-Computer configurations; this is materialized in a taxonomy that chapter 4 describes. After we sampled a small group of case studies, we analysed how they were classified according to our system, and noted which branches were still undeveloped; this served to guide our own research, highlighting the unexplored path we ended up investing in. Following this, chapter 3 details the research problem we aimed to solve, our positioning and research methodology, as well as an outline of the research's work, meant to help guide readers in terms of the experimental procedures of this thesis.

To fully grasp the design of the AGE solution, especially its ethical, theoretical and empirical foundations, read chapter 5. Beyond a design description, the chapter also includes the proof of concept experiment, where a video game's experience was optimized. Then, in chapter 6 are specified all the tasks carried out in order to arrive at a fully fledged Human-computer interface for the AGE tool, including two Participatory Design sessions where game designers paper prototyped it, and doing so, allowed us to better understand its limitations and how it would eventually work in a design process.

Closing up the experimental side of this research, chapter 7 narrates a case study where one game designer used AGE in a design process, allowing us to study its strengths and weaknesses, as well as its appropriation in a real design process. Concluding the thesis, the final chapter 8 reflects on what was learned and contributed to the scientific community, and what remains ahead for this research project.

Chapter 2

Literature Review

To understand how game design and production can be improved with advances in research, we must first look at its state of the art. This chapter is divided in two sections. In the first, section 2.1, we cover how game design and production works today, the importance of player experience for its practice, and how to evaluate and model player experience. This lays the ground for readers to understand the *status quo* of this design discipline, and what future avenues can be explored by researchers to improve it.

In section 2.2, we analyse Procedural Content Generation, the semi-automatic creation of video game content through computational means. Besides clarifying its definition, listing its goals and a number of case studies, we also analyse how to frame them as tools for potential improvement of game design and production. Throughout the chapter, in 'Discussion' subsections, we will present a point of view analysis on these topics, as a way to clarify our positioning.

2.1 Video Games and Player Experience

In this section we start by identifying some of the challenges faced by the video game industry's current production practices – subsection 2.1.1; these will be the issues we aimed alleviating through our research. We present the notion of user experience in subsection 2.1.2, by justifying the concept's enormous operative value in game design activities. Following this is an overview of methods for evaluating a person's experience 2.1.3 – by affording data on specific views of user experience, these methods can help bridge the gap between created artefacts and experienced artefacts, informing designers on how their creations are effectively appropriated by users. To conclude, several experience models created for the design and analysis of video games are explored in section 2.1.4.

2.1.1 Game Design Praxis

One major preoccupation in video game design and production revolves around the need to validate if a given video game fulfils its objectives. Conventional industry video game production involves a long, waterfall-like process with several phases: Concept, Prototype, Pitch, Greenlight, Pre-production, Production, Quality Assurance and Final Gold Master Irish (2005). Besides presenting the initial concept in managerial meetings to garner funding and establish development teams (a process that undoubtedly leads to changes in design, either done in anticipation of these meetings or as consequence of feedback obtained during them), as soon as the concept has been locked, it is prototyped and tested by users, so as to validate it.

Further downstream, there is also a cycle of iterative improvement of all finished design material; this refinement can occur either in Pre-Production leading to a fully fledged prototype, or during Production, leading to the final release candidate (Irish, 2005). Either way, there is a substantial effort in evaluation, testing and refinement of the product even before the Quality Assurance starts near the end of the process.

The recurrence of evaluation and iteration upon existing designs is all the more true in agile game development approaches (Keith, 2010), which iterate shorter cycles of design, implementation and quality assurance tasks. According to Schell (2008), video game testing procedures go from focus groups (probing potential users on how attractive they find given features of a game), QA testing (focused on finding technical issues, usually geared to iron out bugs), Usability Testing (checking if interface and systems are intuitive and easy to use for players), and Play-testing (where players play the game and provide feedback on player experience).

To Fullerton, Swain, and Hoffman (2008), a game's success depends on how innovative and emotionally engaging its mediated player experience is, and therefore game designers focus' should be exclusively on this pursuit. Game design and development processes should be what they call 'playcentric', i.e., focused on delivering a specific player experience, and inviting feedback from players from the beginning of the process, to *"judge the success of gameplay against their goals for the player experience"*(Fullerton, Swain, and Hoffman, 2008). Achieving the desired player experience is thus a major focus of successful game design; and one which typically requires an iterative process of play-testing and refinement to achieve (Fullerton, Swain, and Hoffman, 2008) (figure 2.1

illustrates this).

These industry testimonials attest to the importance that evaluation procedures have in video game development processes; particularly in regards to evaluating player-experience. We think that, given the specificity of this medium, there is an aggravated need to find accurate methods for evaluating how well a given video game fulfils its goals (whatever they may be, commercial, communicational, expressive, artistic, etc.), and these necessarily go beyond testing procedures borrowed from software development's best practices.

The cause for this phenomenon likely lies in the very medium's constraints: video games are highly interactive digital objects, and therefore, subjects' experiences with them are as much dependent on the object's qualities as on subjects' unpredictable interactions with them, as these actively shape the experienced form subjects end up perceiving.

Subjects' interactions with video games determine how video game experience occurs and therefore how the artefact is consumed. But subject-led interactions, while mediated by objects' qualities are, by definition, subjectively motivated. This means that subjects cultural and psychological backgrounds and context – elements completely external to video game artefacts themselves – impacts what facets of the artefact will be experienced and consumed... not just how the artefact will be interpreted (as in non-interactive media).

Hence, on cause of video game's inherent malleability, subjects' degree of appropriation is broader than less interactive media, and this decreases the predictability of video game's mediated experience. As Arseth (2001) argues, *"the complex nature of [videogame] simulations is such that a result can't be predicted beforehand; it can vary greatly depending on the player's luck, skill and creativity"*. Similarly Hunicke, Leblanc, and Zubek (2004) suggest that *"the difference between games and other entertainment products (such as books, music, movies and plays) is that their consumption is relatively unpredictable. The string of events that occur during gameplay and the outcome of those events are unknown at the time the product is finished"*.

The immense variability and unpredictability of video game artefacts, reinforces the need to accurately examine, during the production of video game artefacts, exactly how players are playing them, and what they think and feel about them, so as to evaluate if these are in accordance with the objectives underlying their creation. Evaluating player experience is a rather important step

of game design and development, and one which consumes a great deal of effort and resources. If one wants to improve game design and production, then this is the crucial problem that needs solving... the question becomes, how?

2.1.2 Artefact and Experience

In the previous section, we described how during game design and production there is a significant focus placed on achieving a desired form of player-experience. But just what is an experience?

At lower-income levels, where basic needs satisfaction is difficult to achieve, increased income can have a strong effect on individual well-being; however, once basic needs are fulfilled, higher income values fail to achieve significant increases in well-being, unless income can satiate higher-order psychological needs (Howell and Hill, 2009). Studies have come to show that experiential purchases (i.e. holiday travel, dining, concert attending) can better lead to increased well being, and are more apt at fulfilling high-order needs than material purchases (i.e. jewellery, clothing, TV) (Howell and Hill, 2009; Hassenzahl, 2011). Sensing this, designers are now shifting from product design to experience design, moving design from the creation of artefacts (with material properties) into the realm of creating experiences (with subjective properties) (Hassenzahl, 2011). If design is, as Krippendorff (1989) stated, *"making sense (of things)"*, then experience design is a step forward in that realization, moving beyond both functionality and the aesthetics of things, to aesthetics of experiences, using artefacts not as an end in of themselves, but as means for conveying an experience (Hassenzahl, 2011).

Despite growing interest in understanding what experience is, the answers are diffuse and ambiguous (McCarthy and Wright, 2004). An experience is

> *"an episode, a chunk of time that one went through [...] sights and sounds, feelings and thoughts, motives and actions [...] closely knitted together, stored in memory, labelled, relived and communicated to others, [...] a story, emerging from the dialogue of a person with her or his world through action"*(Hassenzahl, 2011).

McCarthy and Wright (2004) form a more comprehensive picture starting from a pragmatist perspective,

> *"experience can be seen as the irreducible totality of people acting, sensing, thinking, feeling, and making meaning in a setting, including their*

> *perception and sensation of their own actions."*(McCarthy and Wright, 2004)

An experience is then an inherently subjective collection of moments identified by an internal or external principle of unity, e.g., the experience of «riding a bicycle for the first time», «learning that my dog died», «reading a Shakespeare play» or «talking with friends about my workday». The unifying principle gives a name to the experience, attributing it a date, a place - a *context* and maybe even some semantic underlying to how the experience is shaped.

The moments which make up that experience can be succinctly described as *everything* the subject, the *"I"* which lived the experience, recollects sensing, feeling and thinking. Every cognitive whole composed out of thoughts, emotions and sensual input which has been retained by the subject is an inherent part of that experience. Because not all gets processed or stored by the subject, and because our memory is ever changing – with previously recollected moments being forgot, new moments being recollected (both real and imaginary) and new interpretations and semantic attributes being attached to those moments – an experience is, by definition, not only subjective, but ever-changing with time.

Looking at artefacts (such as video games) from the lens of experience changes how one thinks about them. An artefact gains a new frame of understanding, no longer understood as a physical object, with material features we can measure with a ruler, but as a context that enables subjects to experience *something*, and that *something* needs to be dissected in order to understand the artefact's qualities and value. To illustrate this twist in focus, think of a poem: instead of describing the printed words of a poem – their meter, rhyme, cadence, style – we look to how these elements are read, and in so doing, affect subjects' emotions and thoughts.

From these definitions and descriptions, a number of problematic properties of experience can be readily identified. First is its partially subjective character – for an experience is not something that can be objectively measured or analysed, as it is a personal construct, existing only inside a person's own brain and self and perceived self. This immediately makes clear that if one is to fully understand how experience is processed we need to have insight from subjects themselves.

In a more practical perspective, user-experience can be defined as *"all aspects of someone's interaction with a product, application, or system"*(Tullis and Albert,

2008). But regarding user-experience from this more behaviourist point of view – reducing an experience to subject's actions within it – implies disregarding, to a certain extent, the subtler subjective and semantic qualities of experience. The advantage here, of course, is making an experience measurable and quantifiable (Tullis and Albert, 2008).

The second problem lies in its ever changing nature – studying in detail something that is not constant, that lives and breathes as its subject does, is difficult. Memories of the experience shape it, determine its essence, which aspects get amplified and granted meaning, and which get dumped or rejected as if they had never happened.

Finally, one must consider its breadth – an experience is thought, emotion, senses and behaviour spread across time and space, all these in one whole made unique. To dissect such dimensions, or threads, advanced tools and methods are needed, covering as many of these dimensions as possible. Its holistic quality will surely elude analysis, but our understanding of the phenomenon is bound to improve the more we understand these different strands.

In general, design activities have commonly been understood from a communications theory point of view, as if the creation of artefacts were a means for designers to directly convey messages to users; the problem with this understanding being that it severely downplays users' role in appropriation and interpretation of the artefact (Crilly, Maier, and Clarkson, 2008). Indeed, the process is vastly more complicated, even more so in the context of interactive artefacts such as video games, where player interaction can extend to high degrees of reconfiguration of artefacts form.

Social and psychological contexts – expectations, motivations, needs, knowledge base, beliefs – impact severely how the artefact is interacted with, and its consequent interpretation (see figure 2.2). Designers then are burdened with not only with envisioning how artefacts are intended to be appropriated and interpreted, but also all the ways in which they *can* be. Effort then can be made in minimizing misappropriations, and increasing the likelihood that users engage artefacts in ways that are meaningful to them, and which simultaneously respect, at some level, what authors initially intended, including the level of plasticity and permeability to interpretations of the end-artefact.

In that realm, understanding users' experience and how artefacts mediate specific experiences becomes the focus of designers work. If we take this into

consideration, we see that a major challenge in the design process, and moreover in the game design process (in lieu of the medium's multidimensional, interactive character), is to offer designers glimpses of how users interact, perceive and interpret their artefacts.

Such is the focus of the 'user experience' field; its aim being to study how users interact with artefacts, so as to provide meaningful data to all those who are involved in their production (Tullis and Albert, 2008). This data is of the most importance when it comes to obtaining systematic, credible information on the nature of users perception of artefacts, more so in market contexts where the key to a successful product is understanding and catering to consumers.

Once one understands how complex and multifaceted human experience is, it starts to dawn the massive undertaking that is understanding it, even when in reference to a simple context, as is the case of subjects' interaction with a video game artefact. How can this broad phenomenon be measured, dissected and then analysed in order to evaluate it in accordance to some supposition of what it should be? What technologies and methods can be employed in this regard, and how can we model data from such methods in a way that provides meaningful feedback to designers?

2.1.3 Experience Evaluation Methods

Once the concept of experience has been contemplated, a new question is posed: how can experience be scrutinized? What follows is a collection of the major families of tools that can be used in order to probe player-experience.

Self Reporting

Self-reports are the most simple, cheap and direct method of obtaining data pertaining to user experience – it is merely a question of asking subjects how they *perceive* the object. Several different methodologies can been employed in order to proceed to game experience evaluation, mostly diverging in terms of which psychological theory they are grounded on, but also in terms of which qualitative variables to measure; examples: emotion (Lazzaro, 2004; Zagalo, Torres, and Branco, 2005), cognitive psychology (Piselli, Claypool, and Doyle, 2009), flow (Sweetser and Wyeth, 2005) and self-determination (Rigby and Ryan, 2007) theories.

Self-reporting can take many forms, it can be achieved by oral interview, written questionnaire or, in more automated fashion, through electronic surveys,

typically post-experience (Isbister et al., 2006; Tullis and Albert, 2008). Alternatively, questions can be posed mid-experience, or in interviews; subjects can also be shown records of their own experience and asked to comment. Despite this, the two best times for collecting self-reported data are considered to be post-task and post-study; the former gives insight into each tasks specific issues, the latter given a broader overview of the full product (Tullis and Albert, 2008).

Subject reports are still today the most widespread means for evaluating affective experience for in user-product design (Jenkins, Brown, and Rutterford, 2009). They provide the only means we have of directly analysing (the subjective interpretation of) all dimensions of an experience – temporal, stimuli, cognition and emotion, and memory. In some cases this tends to render other data redundant, for if a user is happy with a given artefact or product, then that may be the only thing that matters (Tullis and Albert, 2008).

However, subjective interpretation can skew the usefulness of data obtained via this means. Subjects rely on their memory of the events (experiential memory) when filling out forms, therefore assessing their own biased recollection of felt emotions as opposed to their actual feeling (Ekman, 1999; Dillon et al., 2000). Further limiting the usefulness of these methods, the *"act of repeated measurement can potentially alter the emotional response itself"* (Rottenberg, Ray, and Gross, 2007), impeding possibilities of continuous inquiries to subjects for greater time-definition in trying to characterize emotions. The same is arguably true for other experiential qualities.

Subjective reports also present a wealth of problems regarding their use as a persons' experience assessment method. Results are dependent on subjects' *"perception of the experience and its context and [. . .] ability to interpret and express their feelings"* (Jenkins, Brown, and Rutterford, 2009). This makes them unreliable, with a strong possibility that subjects respond to the test with the reply they think would make them look better in the eyes of others (the researchers), the 'social desirability bias' (Tullis and Albert, 2008; Gross and Levenson, 1995). Questionings, be they verbal, written or otherwise, influence on subjects replies, as they are inevitably conditioned by social constraints on how to answer.

In consumer research, direct questioning has been noted for its relevant number of arising issues concerning users incapacity to properly express their needs (Vankleef, Vantrijp, and Luning, 2005). Some research supports that when subjects have to rationalize their responses to questions, their answers become

impaired (Wilson and Schooler, 1991). Thus, characterizing video game experience based (only) on self assessment reports overshadows many oscillations that occur in a subject's state during different moments in the experience.

Physiological Data

Computer science can now offer alternatives to subjective reports for assessing a person's cognitive and affective state. Emotion, for instance, is not only comprised of a subjective experience but also a number of physiological correlates (Gross, 1999), and these physiological changes can potentially be measured and analysed so as to recognize emotion, as well as other psychological variables.

Measurements of several types of biological signals, from electroencephalography to heart rate activity, skin conductance, temperature, etc. have been used to assess cognitive and emotional states (Ko, Yang, and Sim, 2009; Jenkins, Brown, and Rutterford, 2009; Takahashi, 2004; Khalili and Moradi, 2008; Li and Lu, 2009; Horlings, 2008; Chanel et al., 2006; Haag et al., 2004). Some of these systems show promising results, with correct classification classification accuracy getting closer and closer to 100%; however, research into automatic detection is still in its infancy, with results lacking off-laboratory validation. Pattern recognition systems need vast amounts of data in order to produce general classification models, and right now, as far as we could find, there is no accepted database of cognitive and emotional patterns. There is also no complete and consensual model of cognitive and emotional states. In fact, there is even debate on how to better model emotion and its multi-faceted nature (Frijda, 2008).

The relationship between these bodily processes and cognitive aspects of experience is not fully understood by currently existing theories and methods. Furthermore, there is a case to consider in the difficulty in using these methods on location; while signal capturing devices are getting smaller and smaller, they still tend to be unwieldy, lacking portability, ease of use and wide-spread availability, limiting their wide-spread use in analysis of data from video game deployment.

Game-play Metrics

Game-play Metrics are a recent method used in game-play analysis. Metrics systems track UIEs (user interaction events), producing detailed logs of all actions which users take part in a game (Kim et al., 2008). They automatically

record user behaviour to track specific metrics of interest (e.g. time taken to carry out a specific task), without the need for time consuming ethnographic studies, audio-visual recording analyses or other hand-coded methods (Kim et al., 2008; Drachen and Canossa, 2009). Data can consist of 'low-level' metrics (e.g. keystrokes) or 'high-level' (e.g. player position in the virtual world) (Tychsen and Canossa, 2008), providing several tangible benefits (Drachen and Canossa, 2009), namely:

- high detail, quantitative data on player behaviour,
- an objective method for visualizing and analysing game-session data,
- detailed feedback on game design and mechanics,
- help track location of game problems and helps evaluating fixes,
- customizable detail of analysis,
- easily supplemented by other existing methods for user experience testing and bug-tracking (since data for these purposes can be collected simultaneously).

The focus is on dealing with issues that emerge from game production processes, giving stake holders access to objective data on how the game ends up being played, namely: how players traverse and observe levels, which items, weapons and classes they are more prone to interact with, how often they die in specific level areas, which type of play behaviour is employed, which team-tactics emerge in on-line spaces, amongst many others (Kim et al., 2008; Drachen and Canossa, 2009; Tychsen and Canossa, 2008; Hoobler, Humphreys, and Agrawala, 2004; Chittaro, Ranon, and Ieronutti, 2006). In turn, this information allows for accurate fine-tuning of different aspects of the experience, guaranteeing that designers' original vision is preserved and increasing the probability of commercial success (Kim et al., 2008; Drachen and Canossa, 2009). Furthermore, it allows for constant improvement of video games even after launch, as metric systems can be deployed with titles, communicating data internet services (Kim et al., 2008).

One major issue arises in the transition from the 'what' to the 'why' – metric systems produce large amounts of behavioural data, but do not provide context for their interpretation – which is why metric systems are paired with other methods and then analysed by experts to properly assess the causes and motivations underlying events (Kim et al., 2008; Drachen and Canossa, 2009). In fact, Nacke et al. (2009) argue that to provide a complete picture of game-play experience, several methods must be employed in tandem, for example using

self-reporting to provide the subjective perspective on events.

Furthermore, there is the problem of how data can be processed and presented to designers so as to become useful. Given the large quantities of data generated whilst in play, great effort has to be put in making the right data available in the right format, with considerable research being made on visualization tools for that end (Börner and Penumarthy, 2003; Chittaro, Ranon, and Ieronutti, 2006; Hoobler, Humphreys, and Agrawala, 2004). And this is only in terms of how data can be made meaningful; the following problem being how knowledge of game design issues can be converted into improved designs that solve these same issues. As of now, designers must explore the data and use their know-how and intuition to solve these issues, then await new feedback, and only in the end verify if the problem has indeed been solved.

Discussion

Each experience evaluation method comes with its own set of qualities and issues. Self-reports can give invaluable data on subjective perceptions of the video game artefact, but because they rely too much on subjects' non-expert perspective and lack time-definition, do not have efficacy in assessing, for example, what quality of a game causes a certain evaluation. Game-play metrics provide a high-quantity of high-detail quantitative data on player behaviour, but no framework on how to judge and interpret the data. Bio-metrics will likely in the future provide extensive, objective data on player affect and cognition; but today, the method is still unwieldy to deploy without extensive know-how and cost. All in all, each method requires significant effort in implementing, and only by complementing these approaches can best results be found.

Already today, these methods are employed at different stages of the game design process (Mahlmann et al., 2010), potentially affording designers vast amounts of information on player experience. Game producers then interpret and translate player-experience data into design revisions and artefact production. Precisely for this reason, visualization tools (Chittaro, Ranon, and Ieronutti, 2006) that ease this analytical processes have become assets in making sense of these data streams. While great steps have already been made, further strides could be made if these methods for experience data collection could be framed in the context of tools that besides collecting data, could utilize it automatically for player-experience improvement. We hypothesize that such a move would serve to alleviate the cost of using these methods, and increase their potential impact in bettering game designs.

2.1.4 Player Experience Models

What do we know about player experience and its relationship with game design? Besides analysing experience data-collection methods, in this section we list how said data can be modelled so as to improve designers understanding of player experience. For only with this understanding can player experience data be made indicative of a given game design's quality, and suggestive of subsequent changes needed in order to fulfil its target agenda. The following is an overview of several player experience models which game designers can use as guidance.

Gameflow

Gameflow (Sweetser and Wyeth, 2005) is a *"concise model of enjoyment in games that is structured by flow"*. Flow is the word used to describe a psychological state of mind associated with pure enjoyment, and that has been studied and found in a wide range of different human activities, like games, sports, reading books, watching films, and even work and creative occupations (Csíkszentmihályi, 1990). Sweetser and Wyeth (2005) took the eight major elements needed for activities to convey flow to practitioners – concentration, challenge, skills, control, clear goals, feedback, immersion, and social interaction – and created a model especially conceived for video game activities.

For each element a concrete set of criteria are listed that can be applied to video games, and a set of concrete examples that can distinguish between high-rated and low-rated games. Criteria are expressed in normative guidelines, such as *"games should provide a lot of stimuli from different sources"* or *"games should support competition and cooperation between players"*(Sweetser and Wyeth, 2005). The compelling aspect of this model lies precisely in how it effectively particularizes which key aspects of the video game artefact are needed to convey flow, or enjoyment, to players. Unfortunately, this also means that its scope is reduced when it comes to studying player-experience, as enjoyment is but a part of a larger whole.

Four Keys to More Emotion in Games Without Story

In a similar vein, a research study by Lazzaro (2004), focused on *"why people play games*". Their findings – obtained from analysing 45 subject interviews, game-play video and questionnaires – report there are four keys to more emotion in games. They map fours sets of emotions to four basic game-play types that

elicit them. Hard Fun conveys emotions such as frustration and *fiero*, which arise from meaningful challenges, strategies, and puzzles. Easy Fun triggers player curiosity, wonder, awe, and mystery, through ambiguity, incompleteness and detail in the game-world.

Altered States refers to when games elicit excitement and relief through combination of rich visceral graphic and audio stimuli, intriguing concepts and behaviours. Finally, The People Factor, where through player competition, cooperation, performance, and spectacle, and other forms of social interaction, emotions such as amusement, *schadenfreude* (gloating at a conquered opponent), and *naches* (jubilee face a apprentices' or underling's success) are explored.

Anecdotal examples from the studied sessions are listed as basis for many of the mappings between game-play features and aroused emotions. While they provide a basic vocabulary and grammar for these, their focus on specific instances of games and their qualities in these reduce the study's value. The generalization that is proposed, which takes 4 major types of game-play types and its associated emotions, has the opposite defect: it is too broad and lacking in detail. Between the anecdotal detail of the micro-analysis and the ambiguity of the general model, the operative usefulness of the model is, in our opinion, limited.

Mechanics Dynamics Aesthetics

The MDA framework seeks to relate mechanics, the *"particular components of the game, at the level of data representation and algorithms"*, with dynamics, i.e. *"run-time behavior of the mechanics acting on player inputs and each others' outputs over time"* in order to mediate aesthetics, the *"desirable emotional responses evoked in the player, when she interacts with the game system"* Hunicke, Leblanc, and Zubek (2004).

This three-layered approach helps designers construe how they can impact player dynamics and aesthetics, through an iterative process of design and tuning of mechanics. Besides the three layers, they comprise an admittedly incomplete list of eight types of aesthetics that can then be understood as "fun". These are: Sensation, game as sense-pleasure; Fantasy, game as make-believe; Narrative, game as drama; Challenge, game as obstacle course; Fellowship, game as social framework; Discovery, game as uncharted territory; Expression, game as self-discovery; and Submission, game as pastime.

Despite exemplification of MDA's use in design process, the work's scope is

not developed enough to provide a generalized and detailed systematization of relationships between specific mechanics, dynamics and aesthetics. This model does provide an insightful diagrammatic mode of reasoning for tracing how designers impact video game objects and these in turn impact player experience, which seems valuable in of itself.

Participation-Centred Experience Model

Another three layer-approach is the Game Experience Design Model Centred on Participation (Pereira and Roque, 2012), which distinguishes three operational levels: Intention (game-play experience as idealized by the designers), Artefact (video game object qualities that can mediate player participation and experience) and Participation (player actions and experience as felt by players). Though in this model the artefact view is analogous with Hunicke, Leblanc, and Zubek (2004)'s Mechanics, there is no separation between behaviour and subjective experience, as these are conflated into the term Participation. More so, designer intention for player-experience is herein focused upon, with the design process implicitly defined as a search for harmonious balancing of the three operational levels.

Besides the three-pronged distinction, the model by Pereira and Roque (2012), enunciates a list of six different participation dimensions that were synthesized and expanded upon notions from different authors: Playfulness, *"free, informal, and unstructured participation"*; Challenge, *"structured participation, of a formal challenge, or according to a proposed target"*; Sociability, *"social participation, of establishing relationships between players"*; Embodiment, *"physical participation, both virtual and actual"*; Sensemaking, *"significant participation, of creation of meaning"*; Sensoriality (sic) *"multisensory involvement"* (Pereira and Roque, 2012).

Each type of participation targets experience dimensions very clearly. Because of the predominant conceptual role of the notion of Participation, the *"interpretation of the context of the game and how the player acts in it"* (Pereira and Roque, 2012), there is a strong emphasis on both the subjective and behavioural dimensions of experience. Playfulness and Challenge refer to dialectically opposed forms of player behaviour present in traditional games (Caillois, 2001); sociability refers to all types of social human behavior; embodiment and sensoriality refer to physical and sensual aspects of the experience (which means behaviour is indirectly factored, as it affects our bodily expression), and sense-making refers more directly to the cognitive and emotional experiential dimensions.

Unlike the previous model, this one goes further by forwarding designer-oriented guidelines to operate the model, both in terms of conceptualizing video games with this model (Pereira and Roque, 2013b) and evaluating player-experiences with game-play metrics (Pereira and Roque, 2013a). The latter is particularly interesting, as it uses objective and concise metrics to measure if specific participation dimensions are being explored as intended by designers. Despite these advances, there is no single generalized empirical or theoretical corpus detailing how elements in the different views – Intention, Artefact, Participation – relate.

Fundamental Components of the Gameplay Experience

Another concept that has been employed to conceptualize and value game-play experience is that of immersion. Ermi and Mayra (2005) propose the SCI model that focuses on three types of Immersion – Sensory, Challenge and Imaginative –, derived from analysis of self-reports. Their assessment is that *"immersion is a multi-faceted phenomenon with different aspects that can appear and be emphasized differently in the individual cases of different games and players"*(Ermi and Mayra, 2005).

Their model is, admittedly, not comprehensive; instead it is a heuristic representation of the key dimensions that structure game-play experience. The intention is to guide attention to the complex dynamics of between game and player, with a focus on the consciousness structured by this interplay. Three types of immersion are described.

Sensory Immersion happens when the game's audiovisual stimuli overpower *"sensory information coming from the real world, and the player becomes entirely focused on the game world and its stimuli"*. Challenge-based Immersion arises when *"one is able to achieve a satisfying balance of challenges and abilities"*. Finally, if *"one becomes absorbed with the stories and the world, or begins to feel or identify with a game character"*, Imaginative Immersion is at work.

A questionnaire for subjects to identify these immersion dimensions was then constructed iteratively, arriving at a set of eighteen 5-point Likert scales that served to validate the model. Analysing replies to resulting self-reports allowed the authors to identify the preponderance of the three types of immersion in several commercial video games. Note once again, these three basic types are similar to aspects present in (Pereira and Roque, 2012; Hunicke, Leblanc, and Zubek, 2004; Lazzaro, 2004; Sweetser and Wyeth, 2005), and there is no identification of object qualities that can impact these experiential dimensions.

Reframing the Concept of Immersion

Another model by Calleja (2007) (based on both literature and player and designer inquiries in respect to MMO games), uses the metaphor of incorporation as a refinement of immersion, i.e., *"internalization of the digital environment that makes it present to the participant's consciousness as a domain for exerting agency while simultaneously being present to others within it through the figure of the avatar"*. Incorporation is divided into two phases of engagement with video games, the Macro-involvement, which regards the *"motivation to initiate and sustain engagement with digital games through the long term"* and Micro-involvement, *"engagement that occurs during the game-session"* (Calleja, 2007).

The two phases are seen through six frames of involvement which are refrains on already covered *motifs*: Affective (games' emotional involvement, engaging players in different moods), Tactical (*"satisfaction that results from working towards and reaching goals"*), Spatial (concerning the attractions of exploration and discovery), Performative (draw exerted by the ability to act within the virtual environment), Shared (*"all aspects of communication with and relation to other agents in the game-world"*) and Narrative (involvement by participation in memorable events and stories).

A Tool for Characterizing the Experience of Play

Costello and Edmonds (2009) propose a monolithic model for experience that does not problematize divergences between object and subject. It does offer thirteen dimensions of pleasure in a player experience that work as a state of the art summary of the field's more widely accepted theories. The pleasures and the actions that convey them are: Creation, *"having the power to create something"*; Exploration, *"exploring a situation"*; Discovery, *"making a discovery"*; Difficulty, *"having to develop a skill"* or exercising it; Competition, *"trying to achieve a defined goal"*; Danger, *"feeling scared"*; Captivation, *"feeling mesmerized or spellbound by something"*; Sensation, *"feeling of any physical action (...) such as touch, body movements, hearing, vocalising, etc."*; Sympathy *"sharing emotional or physical feelings with something"*; Simulation *"perceiving a copy or representation of something from real life"*; Fantasy, *"perceiving a fantastical creation of the imagination"*; Camaraderie, *"developing a sense of friendship, fellowship or intimacy with someone"*; and Subversion, *"breaking rules or of seeing others break them"* (Costello and Edmonds, 2009).

Note that the thirteen pleasure dimensions are increasingly detailed variations of experience topics which were conflated into wider categories in, for

example, Pereira and Roque (2012). The Costello and Edmonds (2009) model, like previous alternatives, does not detail any empirical support, nor offer any objective structured mapping between video game artefact qualities and player-experience.

Discussion

2001 was considered *"Year One of Computer Game Studies as an emerging, viable, international, academic field"*; the year in which the first *"academic, peer-reviewed journal dedicated to computer game studies"* was published and *"the first international scholarly conference on computer games"* was held (Arseth, 2001), signalling that research in the game design field is still young.

Our conclusions from surveying the diverse models of player experience are that, one, only a subset of the wider scope of player experience has been the subject of intense scrutiny, namely, the positive aspects of player experience, and two, that none of these models can establish a sufficiently vast, robust method for identifying and predicting how game-qualities impact player experience.

The Issue with Fun

There is a particular incidence on the positive, desirable aspects of player experience (either expressed in terms of 'fun', 'flow', 'immersion', 'pleasure', 'incorporation'), as opposed to a view that seeks to explain player-experience without preconceived judgements. We suggest that these models view experience from the exclusive mindset of the player – what he desires, aims to feel and act-out. These considerations' value notwithstanding, player needs are only a part of the phenomenon of player experience; perhaps the most important in terms of commercial and marketing goals, but a part nonetheless.

Furthermore, core concepts in this space are not supported by consensual models, and even the ubiquitous fun has no consensual, concrete definition (Schell, 2008). Though surely not a dominant view, some research even goes against the current; for example, evidence against 'flow' has been discovered in empirical studies (Lankveld et al., 2009); and immersion has theoretical alternatives (Tavinor, 2009).

Several benefits have been studied and verified in relation to user-centred design philosophies, namely in terms of increased system quality thanks to more accurate requirements, heightened production efficiency, improved user acceptance, amongst others (Kujala, 2003). But we share with other researchers

(Steen, Kuijt-Evers, and Klok, 2007) reasonable concerns over how this user-focused mentality can impair innovation, creativity and even quality (Van Der Panne and Beers, 2003; Heinbokel et al., 1996).

Users can be poor judges of their own emotions and feelings, for example, mistaking strange, unfamiliar or unconventional new experiences as bad or displeasing (Gladwell, 2005). Even proponents of a move towards user-centered design are careful in how they frame it, maintaining a focus on designer's intervening in the process, *"Designers should take an active role in user involvement. (. . .) Users are not just asked to about their needs, but the analysts try to understand their behavior and the future context of use."*(Kujala, 2003)

Indeed, user-centred design presupposes designers to enter into meaningful dialogue with users, in such a way that knowledge is shared from both ends so that design decisions are rightfully informed in collaborative manner (Steen, Kuijt-Evers, and Klok, 2007), never dictated exclusively by user needs. This collaboration can, in fact, be regarded as a continuum of diametrical opposition in terms of approaches, determined by which knowledge is privileged during the process, ranging from 'pro-active user involvement' to 'reactive user involvement' (Steen, Kuijt-Evers, and Klok, 2007).

To illustrate the potential issues in letting user preference dictate design decisions, we forward the 'Pepsi Challenge' story (Gladwell, 2005). The 'Pepsi Challenge' was a TV commercial where coke drinkers were asked to sip and rate their preference of two different cokes Coca-Cola and Pepsi. Results showed subjects preferring Pepsi by a significant margin. Coca-Cola confirmed the effect in their own research, and created 'New Coke', a sweeter drink that beat Pepsi in the blind sip-test. Despite users preference in the test, 'New Coke' was a commercial failure, eventually forcing Coca-Cola to go back to their original formula, which was and so remained market leader.

The discrepancy between what research was suggesting and actual market reaction was result of two factors (Gladwell, 2005). One, the sip-test was inherently flawed measuring reactions out of the adequate context: subjects were not asked to rate preference after comfortably enjoying an entire can or bottle of coke in their homes, but instead asked to do so after only a small sip in a street. This then justified the preference for the sweeter, yet cloyed drink; had there been more time, the initial sugar rush would have lost its initial positive valence, and would have been understood as a negative quality of the drink. And two, the blind nature of the test excluded sensation transference, ignoring

"all of the unconscious associations we have of the brand, the image, the can, and even the unmistakable red of the logo" (Gladwell, 2005).

In video game design and research, user-preference is usually understood under the notion of 'fun'. It is, in a way, video games' own *sugar* – the quality that eclipses all others. Fun is, quite literally, everywhere you look. Game Design books address it with strong emphasis, dedicating whole sections and chapters to its understanding and pursuit (Fullerton, Swain, and Hoffman, 2008; Koster, 2005; Bartle, 2004; Schell, 2008), or with minor references (Juul, 2005; Brathwaite and Schreiber, 2009), and it is always justified as being the sought out quality in players' experience. We question the sense of having an entire creative medium seeking the same illusive quality, or perhaps the different qualities under a single name. Even though it is unquestionable that fun is the word most commonly used by players to determine their preference for a specific video game. Anyone in the field of video games would be hard pressed not finding a reference to the three-letter word in practically every article that concerns video games in any way.

But is fun really all players want? Randy Smith, lead game designer, eloquently exposed the argument in his magazine column:

> *"Because games are supposed to be fun (...) We've brilliantly succeeded in eliminating the interstitials, stripping away everything but fun. And what's with fun? Schindler's List is a valuable film, but it's not especially fun. (...) How did we become the artform that absolutely has to be all about fun? Remember when graphic novels were all about superheroes and cartoon animals? Was that so great?"*(Smith, 2008)

The curious thing is that, whilst there is an attempt to understand what the word means (see, as an example, (Vorderer, Hartmann, and Klimmt, 2003) or (Lazzaro, 2004)), there is an acknowledgement of its highly ambiguous, undefined nature:

- *"sometimes fun defies analysis"*(Schell, 2008);
- *"it's a somewhat circular definition: Players play so as to have fun, fun being what they aim to feel while playing"*(Bartle, 2004);
- *Unfortunately, fun is one of the most elusive concepts you ever try to pin down"*(Fullerton, Swain, and Hoffman, 2008);

Fun is really a catch-all expression, reducing player experience, emotion and satisfaction to a single word, in reality saying practically nothing other than video games are to be entertaining experiences. Fun is construed as being inherently associated with pleasure (Salen and Zimmerman, 2004; Brathwaite and Schreiber, 2009; Schell, 2008), or as in (Chen, 2007), with Csikszentmihaliy's concept of flow (Csíkszentmihályi, 1990) (in itself, flow is a general psychological model for the arousal of pleasure).

Many attempts have been made to further structure what 'fun' is and how it can be elicited. Bartle defined 'fun' in terms of player behaviour, separating it into four player types – achievers, explorers, socializers and killers – each deriving 'fun' from different aspects of on-line game-worlds (Bartle, 2004). Lazzaro (2004), studied how video games' gameplay could be emotionally expressive and concluded there were 4 different emotional keys associated with 4 different types of video game experience, two of which were 'hard fun' and 'easy fun', the first deriving from meaningful challenges (also backed by Vorderer, Hartmann, and Klimmt (2003)), the second from exploration and immersion in the game world and narrative. Fullerton, Swain, and Hoffman (2008) expanded the rationale to other types of fun; and Salen and Zimmerman (2004) covered several other typologies of 'fun' and pleasured, using categorizations based on structural features of the video game.

What is clear from all authors is that 'fun' is everything but a clear concept. Authors avoid defining it precisely, yet are keen on appropriating it to define video games expressive purpose. As an emotional state of being, it is not even mentioned by emotion research experts Ekman (1999), Frijda (2008), and Plutchik (2001). So, when it comes to design authors and scholars and journalists to define what really matters, what a video game can express, the answer is simultaneously elusive and somehow associated with the idea of pleasure.

Besides providing emotional gratification with positive emotions, films can sadden, anger and disgust us (Gross, 1999), and such an emotional vocabulary is accepted as fundamental to the medium. Video games, on the other hand, have been known to have severe constraints in terms of emotional expression, for example being seemingly incapable to elicit sadness in interactive segments, as shown in (Zagalo, Torres, and Branco, 2005; Zagalo, 2009). Because they have such flaws, it seems we continue propelling them unto the future, assuming 'fun' is all there is to their emotional expression.

What seems to be suggested (for example, in Fullerton, Swain, and Hoffman

(2008)) is that if a game is not pleasurable from the beginning and does not continue to give positive emotional feedback, then it is not a video game which can captivate the audience. Simply put, a bad video game. Good video games are fun. Bad video games are frustrating. Or so it is implied. The usage of the word 'fun' – in academia and journalism – does not seem naive to us. It resonates with a mercantile approach to video games, and with a perception of video games as undemanding, effortless activities, where the ratio of effort to pleasure favours the latter. This is, we believe, the linguistic translation of a hedonic view of video game playing, which sees immediate pleasure, enjoyment and comfort ('fun') as the single, ultimate purpose to the activity.

We expect that video game players expectations should further propel these notions when it comes to promote some form of preference in terms of experience. Game designers as well, like other members of the design field, are *"reducing experience to the mere "pleasure due to the feel of the action"* (Hassenzahl, 2011). Experiences can be a way to fulfil greater psychological needs (Hassenzahl, 2011), and do not come to us ready-made, the effort we undertake is a requirement for the quality and meaningfulness which the experience comes to possess (McCarthy and Wright, 2004).

While we think hedonism is a valid point of view with how to frame video games, it must not be the only one. We should also look at video games from an Eudaimonic point of view, according to which *"true happiness is found in the expression of virtue – that is, in doing what is worth doing"*(Ryan and Deci, 2001). Psychology research actually shows that hedonic and eudaimonic pursuits in tandem help lead to a happier, more fulfilling life than just catering to one motivation – hedonic relating more to positive affect and care-freeness, eudaimonia to higher meaning and elevating experiences (Huta and Ryan, 2010).

Self-Determination Theory (SDT), a widely accepted psychological model, posits that three basic psychological needs – competence, autonomy and connectedness – *"as essential for psychological growth (e.g. intrinsic motivation), integrity (e.g. internalization and assimilation of cultural practices), and well-being (e.g. life satisfaction and psychological health), as well as the experiences of vitality and self-congruence"* (Ryan and Deci, 2001). As can easily be deduced, this perspective embodies a strong eudaimonic point of view on self-realization and well-being (Ryan and Deci, 2001). SDT has even been employed as a form of predicting a video game's success in terms of player enjoyment and interest, game ratings, sales and developer loyalty, as opposed to 'fun' oriented methodologies (Rigby

and Ryan, 2007).

As a consequence of the focus on players and their entertainment value, game qualities and game designer's agenda – seen from a distanced perspective, a less player-centric point of view if you will – are significantly under-explored, and that research (like Pereira and Roque (2012) and Hunicke, Leblanc, and Zubek (2004)) is lacking in empirical support and detail. The latter attempts at dissecting the holistic player experience phenomenon – through differing methods, and based on divergent literature and theoretical sources – have lead, perhaps inevitably, to discretionary forms of categorization. Consequently, these prevent any one model from becoming consensual.

There seems to be a clear overlapping experiential qualities between them, merely framed with different metaphors, organized with differing levels of granularity, and/or attributed disparate degrees of prominence. The synthesis work of Costello and Edmonds (2009) illustrates how there seems to be a vivid pattern lurking beneath all these models. And recent research into self-report questionnaires based on existing player experience models shows that (despite there being room for improvement) there is considerable convergence between player responses, with different questionnaire results showing significant statistical correlation between them (Alena Denisova, 2016).

Our positioning is that just as human experience is a complex multi-dimensional phenomenon that is difficult to isolate and analyse, so player experience is bound to be just as rich and complex. This is not to deny the operative quality of these models to guide and instruct game designers on how to achieve certain experiential effects. We merely wish to highlight that these models simply do not provide a general enough view of their underlying phenomenon and/or establish little tangible evidence of their predictive effect, especially in terms of relating how specific video game object qualities influence players' subjective experience.

How to Interpret and Operationalize Player-Experience Data?

The lack of consensual models and guidelines hints at the need of substantial *ad-hoc* work involved in analysing player experience. This is, of course, expected in a design field that is recent and inhabited by very distinct perspectives on what constitutes a successful artefact. However, this effectively shows that there is a possibility here for improvement of game designers' work. Designers have a great deal of player-experience data retrieval methods at their disposal, but lack solid design frameworks to empower their use. Therefore, any tools that can

ease the process of interpretation and operationalization of player-experience data may have a significant impact on game design and production processes.

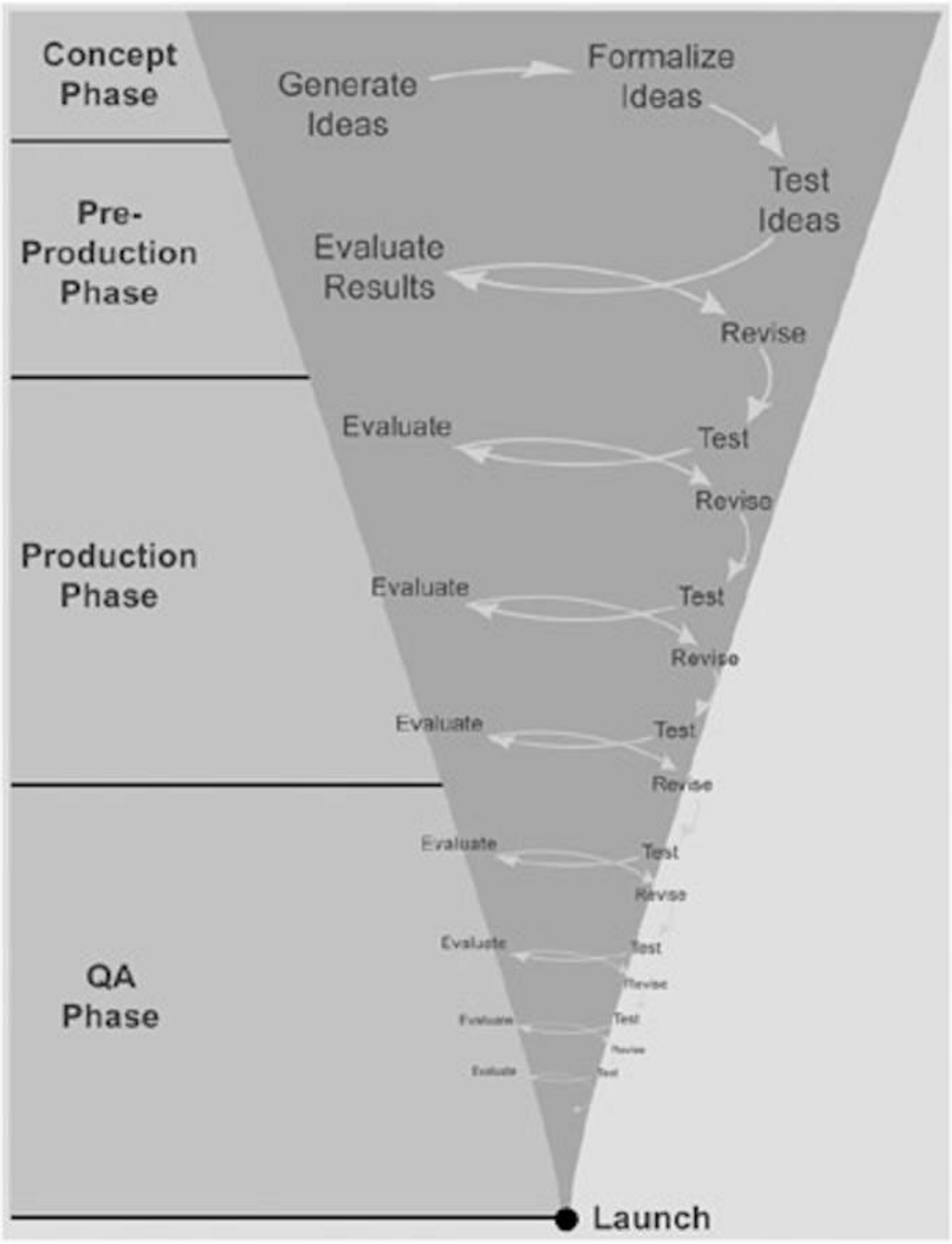

FIGURE 2.1: Iterative game design according to (Fullerton, Swain, and Hoffman, 2008); heed the cyclical nature suggested by the spiral metaphor, and the underlying motto: *"We advise playtesting and iterating your design from the very moment you begin"*.

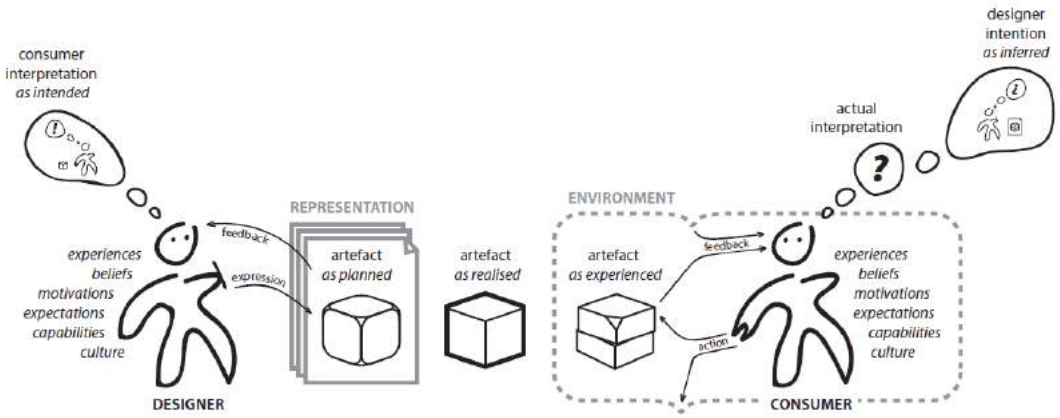

FIGURE 2.2: Design as Communication – Integral Model, Interpersonal View (Crilly, Maier, and Clarkson, 2008). Note the multi-dimensional facets of the artefact which designers should take into account.

	Intention	**Artifact**	**Participation**
Playfulness	exploring, discovering, recreating, customizing	the nature of a player's agency, the variety of interactive elements of the game (objects, characters, actions, etc.)	degree, variety and tendency of exploration
Challenge	overcoming a challenge, creating a strategy, defeating an opponent, mastering a skill	nature of challenges proposed, type of penalties and rewards, intensity and organization of challenges	control, pace, progress, efficiency in performing tasks
Embodiment	physical involvement, physical performance	representation of the physical game world, player's representation on the game world, interpretation of player's movement	control and rhythm of movement, aesthetics of the movement
Sensemaking	interpretation of a role, fantasy, self-expression	theme and underlying narratives, models and representations of phenomena, roles and motives, significant actions	alignment between actions and roles, understanding and or critique of the represented phenomenon
Sensoriality	contemplation, wonder	style, nature of the stimuli, visual and sonic compositions, synesthetic explorations	degree of exposure and responsiveness to stimuli, interaction or engagement with sources
Sociability	competition, cooperation, friendship, identification, recognition	diversity and nature of social interactions and relationships, models of social structures (team, hierarchy, etc)	the intensity and types of interactions between players, affectiveness bonds

FIGURE 2.3: The Participation-centric Player Experience Model (Pereira and Roque, 2012)

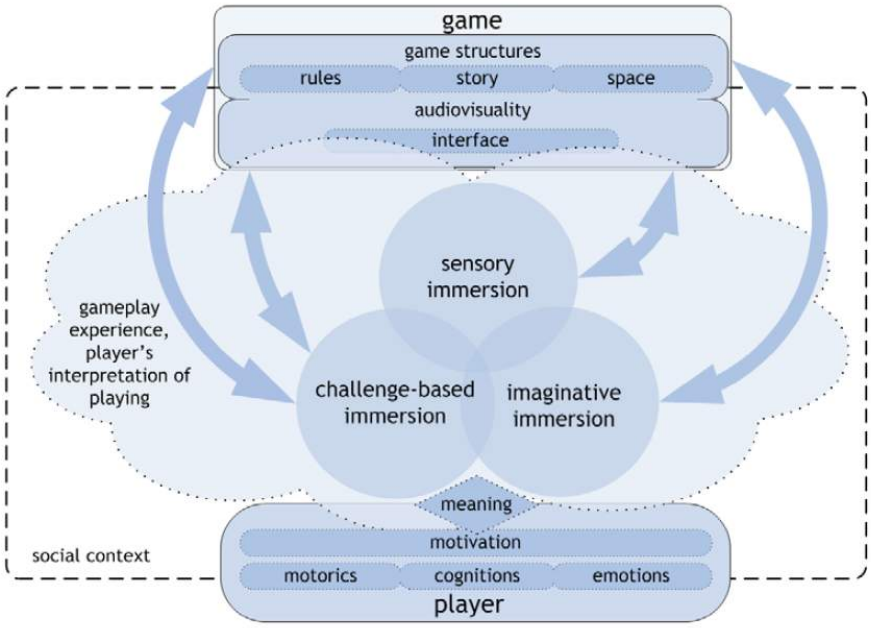

FIGURE 2.4: The Sensory, Challenge and Imaginative Immersion Model (Ermi and Mayra, 2005).

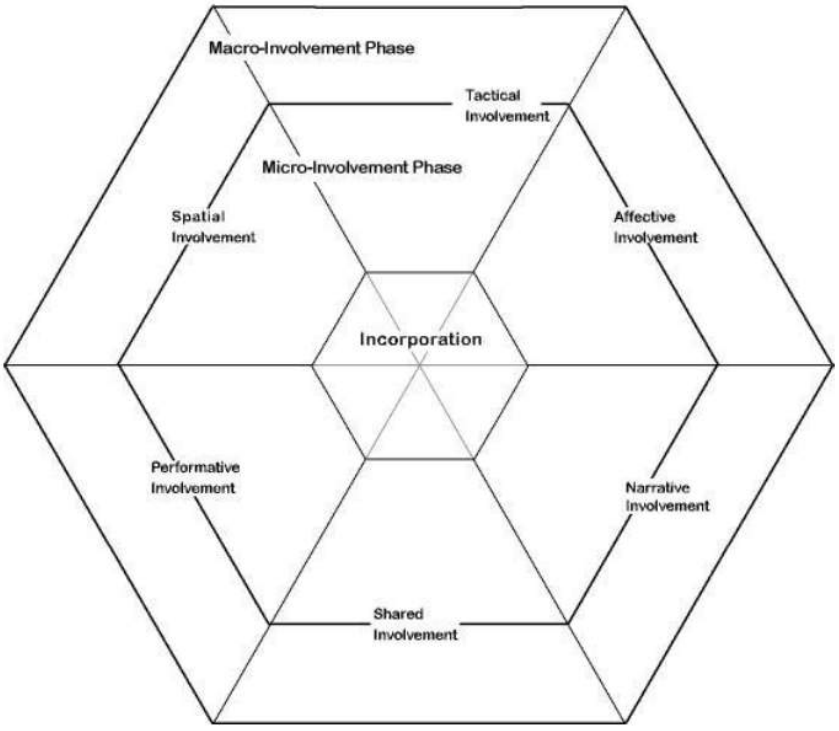

FIGURE 2.5: Incorporation Player-experience Model (Calleja, 2007)

2.2 Procedural Content Generation in Games

Whilst in search of further methods that could better inform game-design in view of improving player-experience, we came across the approach of Experience-Driven Procedural Content Generation (EDPCG for short) (Yannakakis and Togelius, 2011). EDPCG is interesting in the sense that it takes a vast accumulation of scientific knowledge and technology into both experience evaluation methods and artificial intelligence algorithms, and applies it to a computational method that seeks to automatically improve player experience. EDPCG of course, is only a particular type of Procedural Content Generation (PCG), a not-uncommon method used in commercial practices for the creation of game content (level design, art assets, AI-controlled characters) by automatic or semi-automatic algorithmic means (Hendrikx et al., 2013; Yannakakis and Togelius, 2011).

PCG overlaps with Computational Creativity(Cardoso, Veale, and Wiggins, 2009), a sub-discipline of Artificial Intelligence. It is the "philosophy, science and engineering of computational systems which, by taking on particular responsibilities, exhibit behaviours that unbiased observers would deem to be creative", and it aims to *"build and work (...) computational systems that create artefacts and ideas"*(Colton and Wiggins, 2012). Computational Creativity systems are capable of generating non-photorealistic art, music, jokes, stories, mathematics, engineering, among others (Liapis, Yannakakis, and Togelius, 2014), and the cultural value of these artefacts has been validated on several occasions(Colton and Wiggins, 2012). Perhaps herein lies a way for technology to assist game designers – by automating part of the creative act involved in the game design process?

The classic example of use of PCG algorithms is the game Elite (Brabben and Bell, 1984), a space resource management and combat simulator game where, in order to fit a universe (with eight galaxies, 256 stars each) in a limited memory space, each planetary system was compressed into a single seed number (Spufford, 2003). Then, in order to expand, or generate, each solar system, a Fibonacci-like pseudo-random number generator was used to map seeds into a set of pre-coded solar system specifications on the fly, resulting in a combinatorial explosion of different possibilities for players to explore. Specifications covered everything from planetary physical measurements, names to their inhabitant's features (Spufford, 2003).

In Elite (Brabben and Bell, 1984)), PCG was used primarily as a way of circumventing existing memory limits for physical media, while still incorporating

large quantities of content, as *"procedurally represented content can typically be compressed by keeping it 'unexpanded" until needed"* (Togelius et al., 2011). Despite his goal, player experience was impacted in important ways by the game's capacity to deliver vast new worlds, producing a prolonged sense of discovery and novelty that a game with pre-defined worlds would have greater difficulty in achieving. However, because the generator is working with a pool of pre-defined features and numerical statistics, there is a limit to that sense of novelty. Elite (Brabben and Bell, 1984), in this sense, shows both the potential an pitfalls of PCG: it can provide *"permanent originality"* at the risk of exposing *"the underlying sameness of all the star systems, generated as they were from only a handful of varying qualities (...) pink volcanoes would come round again and again"* (Spufford, 2003). The downfall of these techniques is that compared to manually designed content, procedural content has the potential to look dull and repetitive (Tutenel et al., 2009).

2.2.1 Goals

PCG can have several different goals beyond just memory considerations. Hendrikx et al. (2013) highlights PCG as a method for solving issues of burgeoning development costs, by increase of both efficiency and efficacy of content production for video games. PCG is also seen as a method to optimize player experience (Yannakakis and Togelius, 2011; Togelius et al., 2011), and that is the stated goal of most EDPCG methods (see (Pedersen, Togelius, and Yannakakis, 2009; Pedersen, Togelius, and Yannakakis, 2010; Yannakakis and Hallam, 2006; Togelius, Nardi, and Lucas, 2006; Shaker, Yannakakis, and Togelius, 2010)). Experience optimization itself is defined as a way of maximizing players' experiencing of *"fun, (...) satisfaction and entertainment"*(Yannakakis, 2008).

The optimization process implicitly serves the goals of increased efficiency and efficacy in development, but provides two additional (and entangled) benefits, customization and re-playability (Yannakakis and Togelius, 2011; Togelius et al., 2011; Shaker, Yannakakis, and Togelius, 2010; Yannakakis and Hallam, 2006; Yannakakis and Hallam, 2009). The first, refers to PCG's capacity to deliver high quality content tailored to different player types or to the same players while feeling distinct affective states (Togelius et al., 2011); or as Smith et al. (2012) puts it *"a PCG system can be employed as an on-demand game designer, capable of crafting a unique experience for each player"*. The second benefit, re-playability, lies in PCG's potential to continuously deliver novel (or, at least, seemingly

novel) content, increasing long-term re-playability of the same game-product, which in turn increases its commercial value; to Togelius et al. (2011), *"If new content can be generated with sufficient variety in real time then it may become possible to create truly endless games (...) with close to infinite replay value"*.

In her taxonomic overview of PCG, Smith (2014) states a design point of view, and hence was keen on studying how PCG affects players, mediating new forms of player experience that were either difficult or impossible to realize without procedural methods. Despite not being their main focus, Togelius et al. (2011) also recognize the major impact PCG can have in game designs, as *"it might allow the emergence of completely new types of games, with game mechanics built around content generation"*, with *"the breadth of content potentially affected is only beginning to be understood"* (Togelius et al., 2011). If these methods are automatic or at least semi-automatic, the question becomes *"whether computers will ultimately yield designs that compete with human imagination"*(Togelius et al., 2011). The notion that PCG can (partially or completely) replace human authors in creative ventures, is nevertheless shared, to greater or lesser extent by different authors with different positioning.

Considering the current technological landscape, where physical memory is common and digital distribution through fibre networks a commodity, the particular end-use of PCG methods for memory limit circumvention, while not obsolete, seems of much lesser value than it was decades ago. As such, we will refrain from giving it primacy, and avoid addressing it from hence on.

To summarize, we will enumerate PCG intended benefits for game design and production (note these are not mutually exclusive).

1. Effective and Efficient: enable higher quality content production with low-cost methods (Hendrikx et al., 2013);
2. Re-playable: increase game's longevity, by providing ways of automatically generating endless game scenarios (Togelius et al., 2011);
3. Adaptable: allow automatic personalization of content to players' personal needs, play-styles and/or affective states (Yannakakis and Togelius, 2011);
4. Entertaining: optimize players experienced sense of fun (Yannakakis and Togelius, 2011);
5. Expressive: empower game designs that affects players' experience in novel ways (Smith, 2014).

Our own positioning is that PCG can be understood as authoring tools, and can potentially help designers have greater influence over video game's

mediated player experience, be it novel or not – a point we will expand in the next chapters.

2.2.2 Taxonomies

Because PCG is such a wide field where the same algorithms can be used in a myriad of ways, surveying its many diverse trappings seemed important to establish a firm understanding of available alternatives. Because there are several different families and approaches to PCG, there is no distinct form of surveying, cataloguing and classifying methods, so what follows is an overview of several taxonomies in this research field that highlight how PCG works and what methods can be employed in our own research.

PCG for Games: A (Commercial Games) Survey

Hendrikx et al. (2013) proposes a system for classification that distinguishes content type and generation method of each solution. In respect to content type, they present a hierarchical categorization (which is further subdivided): Game Bits, Game Space, Game Systems, Game Scenarios, Game Design, and Derived Content.

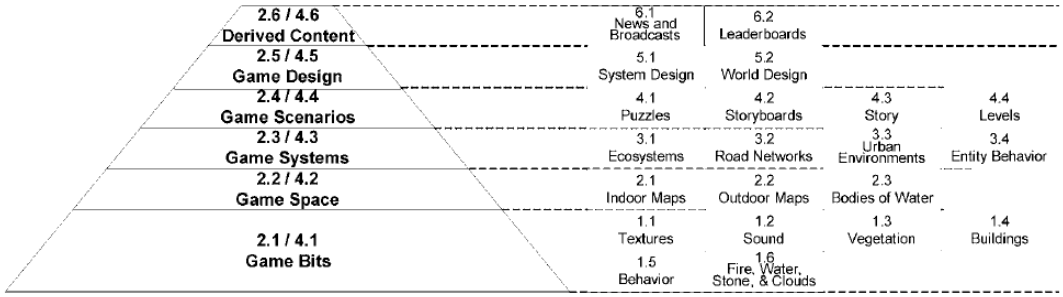

FIGURE 2.6: Content types according to Hendrikx et al. (2013)

Game Bits are elementary units of game content that the player, usually, does not directly engage with; these include cosmetic elements such as textures, sound and vegetation, but also behaviour (i.e. the way objects interact with each other and the environment). Game Space is the spatial environment where play takes place in. Game Systems refer to all complex modelling of real-life phenomena represented in the game, for increased believability and appeal. Game Scenarios describes the way and order of video game events, from story events to levels and puzzles. Game Design refers to game rules and goals, and also the design of the theme, setting, and story (though it is not entirely clear how this sub-content

is conceptually orthogonal to Game Bits, and Game Scenario). Finally, Derived Content is all content that is a side-product of the game world, this includes news and broadcasts that communicate game events and player leader boards.

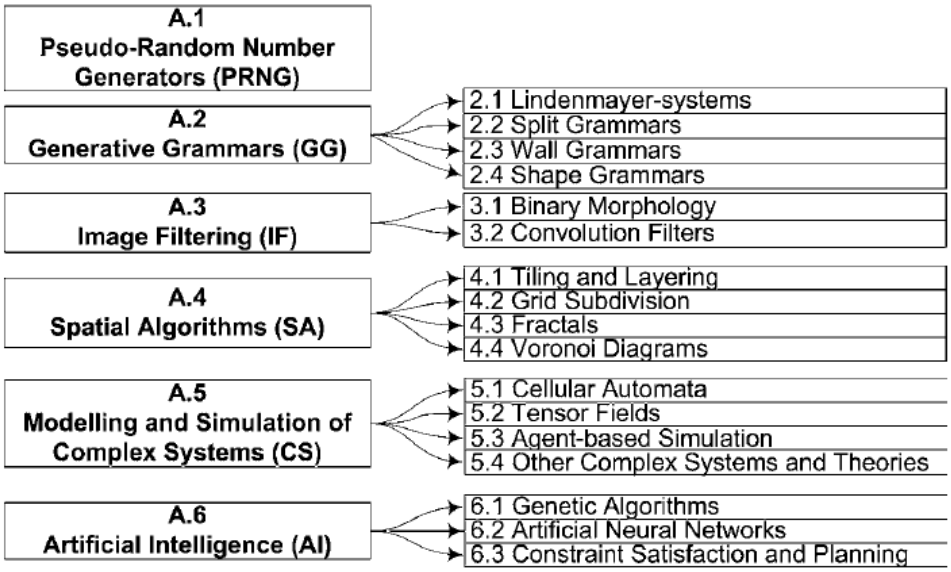

FIGURE 2.7: Families of generation algorithms (Hendrikx et al., 2013)

Hendrikx et al. (2013) then establish six large families of fundamental generation methods used in both literature and development praxis: Pseudo Random Number Generators, Generative Grammars, Image Filtering techniques, Spatial Algorithms, Modelling and Simulation of Complex Systems and Artificial Intelligence methods, with further sub-categorizations to each of these.

Design-centric Analysis of PCG

After over-viewing several PCG games and research projects, Smith (2014) proposed an analytical framework and corresponding vocabulary for understanding the role of PCG in games from a design point of view. Smith starts by listing 5 categories for content generation approaches, based on the procedural aspects of algorithms' generation: Optimization (approaches that *"search for the combination of elements that best fit some criteria"*(Smith, 2014)), Constraint Satisfaction (declarative specification of properties and constraints that define the generated content), Grammar Approaches (content is generated by an expansion of grammar-like rules), Content Selection (selection of experienced content by sampling from a larger databases) and Constructive Generators (generate content by assembling previously authored building blocks).

Besides this categorization, Smith further particularizes her framework by viewing it through the lens of the Mechanics, Dynamics, and Aesthetics (MDA) framework (Hunicke, Leblanc, and Zubek, 2004). Smith's view is that PCG methods can be encompassed as integral part of a game experience, and consequently, aims to study PCG from a design perspective, following the MDA framework. Thus, in regards to the mechanical dimension of PCG (both how the generator works and how it can used), four different aspects are detailed (see figure 2.9).

Building Blocks	**Game Stage**	**Interaction Type**	**Player Experience**
Experiential Chunk Large, human-authored	*Offline* Before game	*None* No human influence	*Indirect* No direct experiential control
Template Computer fills in blanks	*Online* During game	*Parameterized* Indirect, human sets values	*Compositional* Human influences available components
Component Pattern Small, human-authored		*Preference* Human selects good products	*Experiential* Human influences player experience
Subcomponent Internal representation		*Direct Manipulation* Human manipulates product	

FIGURE 2.8: Summary of the four mechanical aspects of PCG, and their potential values (Smith, 2014)

Building Blocks refers to how design knowledge is represented in the generator, therefore impacting what content can be produced. Experiential chunks are large building blocks for content designed by human authors, that can by themselves, be experienced by players. Templates are a generalized form of experiential chunk where human designers leave 'blank' spaces for the computer to fill in.

Component patterns are human designed components that are small enough that by themselves do not greatly impact player experience. Subcomponents, refers to generators that operate at the subcomponent level, using the same building blocks as a human designer would.

Game Stage refers to when generation occurs. On-line means content generation happens during play experience, and Off-line meaning that when the game is played, all content has already been previously generated before a play session begins. Interaction-type refers to how the generator takes (or does not take) into account players' interactions so as to determine what content is created. 'None' refers to a generator that is not interactive, 'Parametrized' one where player can select specific variables used in generation (for example, defining that a map should have a given size), 'Preference' when the generation takes players' preference into account (whether implicit or explicitly declared), and

Direct Manipulation, where the player can actively alter and recreate content that results from generation.

Player experience refers to how player experience is impacted and conceived in the perspective of the procedural content generator. Is generated content geared toward a specific player experience? Can the player affect how the generator impacts player experience? The generator may simply impact player experience without any directing, conceiving or perceiving it (Indirect), it may allow control on how components are created, therefore impacting indirectly the mediated experience (compositional), and Experiential, where the generator seeks to mediate a specific player experience.

Other Mechanics	**Memorization vs. Reaction**	**Strategizing**	**Searching**	**Practicing**	**Interacting**
Core Reliant on PCG	*Memorization* Testing player memory	Player builds strategies for influencing the content generator	Player seeks out new content in a vast world	Player practices game mechanics in new settings	Communities of players discuss differences in game experiences
Partial Framing Partial player experience	*Reaction* Reacting to unforeseen circumstances				
Decorative Not core to game experience					

FIGURE 2.9: Summary of the dynamic behaviors elicited by PCG methods (Smith, 2014)

The dynamic dimension of the MDA framework concerns the behavioural dynamics of players once they are appropriating the game artefact (Hunicke, Leblanc, and Zubek, 2004). Smith (2014) goes on to divide dynamics into 6 sub-categories: Other Mechanics, Memorization versus Reaction, Strategising, Searching, Practising, Interacting. Other Mechanics refers to how the generation impacts player interaction with the game and consequently, player experience. Core means that PCG impacts the core interactions and player experience, therefore making it heavily reliant on how generation works. Partial Framing, as the name so inclines, means that it only has partial influence. Decorative meaning generation only impacts purely decorative elements.

Memorization versus Reaction indicates how, by generating content, a PCG game can impose different behavioural dynamics for the player. Say a game has a set of pre-generated levels, which it can select for him to play, if the form is static, then the player can learn how to memorise the game-play patterns he needs to enact in order to become successful, whereas if levels morphological features are generated on the fly, the player has to become adept at reacting to the new content, instead of just memorizing it. Strategizing refers player's

tendency to play in accordance to strategies that take into consideration how the generator works; for example, if there is some sort of dynamic difficulty adjustment to generated content, player may consciously perform worse so as to get easier content from thereon.

Because generation can increase the scope of game worlds considerably (without human authors' logistic cost), players can feel enticed to explore content in search of new surprises, which is what the 'Searching' dynamic refers to. 'Practising' refers to games that offer generation that enables players to practice game strategies in a variety of different environments. 'Interacting' then points to examples where players interact with each other in order to share game-play experiences, and how generation impacts them, as an example, *'Civilization IV players engage in long discussions about how different map generation options work, how they impact strategies, and why they prefer certain options"*(Smith, 2014).

Finally, when it comes to the Aesthetics, Smith identifies three types of fun PCG can support: fun of Discovery (enjoying the finding new spaces, new environments, new mechanics, etc.), of Challenge (meeting different challenges achieved with novel enemy types and enemy positioning, level structures, etc.), and of Fellowship (players banding together to share strategies on how to tackle procedurally generated challenges), and relates these to the dynamics that can help enhance these positive sentiments.

Search-based Procedural Content Generation

Togelius et al. (2011) distinguish PCG based on five dialectical factors. First, 'Online versus Offline', which distinguishes when generation occurs, whether during game development, or while players are playing games'. Then, 'Necessary versus Optional Content' differs content that is required by players to progress in the game and content which players can avoid (the relevance of this contrast for the authors is that necessary content is obliged to higher standards than that which is optional).

'Random Seeds versus Parameter Vectors' differentiates generators based solely on a random seed from those that can take as *"input multidimensional vector of real-valued parameters that specify the properties of the content"* to be generated, therefore establishing a scale for parametrization with several degrees of control (Togelius et al., 2011).

'Stochastic versus Deterministic' distinguishes random generators from to those with a predictable, consistent output when given the same input. Finally,

'Constructive versus Generate-and-Test' highlights the procedural nature of the generator; whether it constructs its end result by a series of operations and rules that guarantee a minimum degree of correctness and quality, or whether it has a phase of generation and one of testing the content, using some criteria.

Togelius et al. (2011) go further and define 'Search-Based Content Generation', a sub-type of 'Generate-and-Test', where the test function grades content in a real-number scale, using an evaluation function, with the generation process being contingent on that quality measurement so as to produce higher quality content. Finally, they further expand their analysis with three small elaborations on structural components of this particular type of PCG. We are especially interested in their categorization of different types of 'Evaluation Functions', into 'Direct', 'Simulation-based' and 'Interactive' types.

'Direct' refers to any evaluation is based on intrinsic qualities of generated content, and further distinction is then made between data-driven and theory-driven direct evaluation. In *"theory-driven functions, the designer is guided by intuition and/or some qualitative theory of player experience to derive a mapping"* between content features and quality and *'data-driven functions are based on data collected on the effect of various examples of content"*(Togelius et al., 2011).

'Simulation-based' refers to when evaluation is based on game-play data extracted from simulated walkthroughs by artificial agents. Simulations can be static or dynamic; in dynamic variants, the agent changes while playing the game, for example, learning and improving while doing so, and the evaluation function will take that into consideration, while in static variants it remains the same.

Lastly, 'Interactive Evaluation Functions' are those that use human player game-play data, either explicitly, from subject replies to questionnaires and verbal expressions, or implicitly, by measuring behaviour and expressions of affect (such as facial recognition and bio-signals).

Experience Driven Procedural Content Generation

Our greater interest in PCG lies in EDPCG, the family of PCG methods that aim to improve, or optimize, player-experience (Yannakakis and Togelius, 2011). EDPCG is defined as *"generic and effective approach for the optimization of user (player) experience"* with player experience being understood as *"the synthesis of affective patterns elicited and cognitive processes generated during gameplay"* (Yannakakis and Togelius, 2011).

The main concept is to use a data source to create a player experience model (resulting of the interaction of players with the game system), and then use PCG algorithms to generate game content that, according to the model, and a specific value function, optimizes said player experience. Data sources that can be used for the assessment of player experience include all the aforementioned ways of collecting player-experience data: game-play metrics, physiological data, subjective reporting etc., with several alternatives falling in these families. Experience quality is assessed based on player experience data, irrespective of data collection method. Once quality has been assessed, content is represented in computational form, and new game content is generated with a search algorithm. Figure 2.10 shows a schematic that outlines the main modules of the approach.

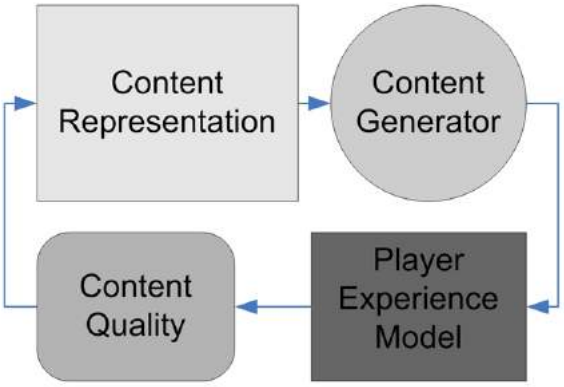

FIGURE 2.10: The 4 phases of Experience Driven Procedural Content Generation (Yannakakis and Togelius, 2011)

EDPCG techniques can be used to: create new games or game-prototypes (Smith, Whitehead, and Mateas, 2010b; Togelius and Schmidhuber, 2009), create and reconfigure levels for existing games (Togelius, Nardi, and Lucas, 2006), generate models of how players tend to perform in a game (Mahlmann et al., 2010; Drachen, Canossa, and Yannakakis, 2009) or which strategies they are more likely to use in competitive gaming (Weber and Mateas, 2009), improve AI routines that are adequate for players or can make the game more interesting (Yannakakis and Hallam, 2004; Yannakakis and Hallam, 2005), and finally, it can be use to balance existing game designs (Lankveld et al., 2009) and adapt a wide range of game and level parameters so as to optimize experience and entertainment (Pedersen, Togelius, and Yannakakis, 2010; Yannakakis and Hallam, 2009; Shaker, Yannakakis, and Togelius, 2010).

After surveying existing examples of EDPCG approaches Yannakakis and Togelius (2011), propose a taxonomy for classification by offering alternative

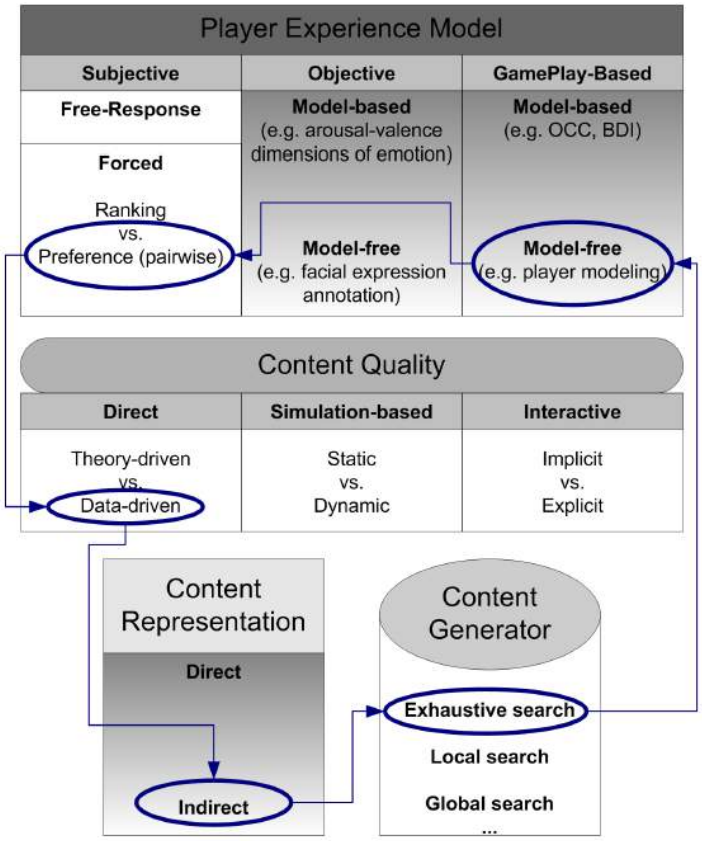

FIGURE 2.11: Yannakakis and Togelius (2011) Experience Driven Procedural Content Generation approach and its taxonomy of methods.

approaches to each of the four components. Experience Models are divided into three basic types that can be used in conjunction: Subjective (based on player self-reporting), Objective (based on objective measurements of player experience, such as bio-signals) and Gameplay-based (where experience is indirectly derived from in-game behaviour). Subjective includes two variants: free-response (when players unrequited expressions are monitored) and forced (questionnaires and other directed means of assessment). Forced self-reports are then divided into ranking, in which players answer questionnaire items in a scaling forms, and preferences, in which they directly compare their experience with different content. Objective Modelling and Gameplay-based subdivide into Model-based approaches (where data is mapped into a model, such as a cognitive or emotion theory) and Model-free (which rely on unknown mappings between player input and emotional/cognitive states, based for example on user-annotations).

Following their previous work (Togelius et al., 2011), Content Quality (as was

the case for Evaluation Functions) is evaluated through either direct, simulation-based, or interactive means, each with the same sub-categorizations as before. Content Representation is divided into direct (genotype size is linearly proportional to the phenotype, with each genotype locale directly mapping to the phenotype) and indirect (genotype maps non-linearly to the phenotype and without the need for proportionality) encodings.

Finally, according to this taxonomy, generation can be exhaustive (all content is generated), local (only generates alternatives surrounding a specific point in content-space) and global (searches the whole content-space, but with no guarantees of completeness). With this taxonomy come several considerations on the pros and cons of each method alternatives.

2.2.3 Survey

We consider PCG algorithms can be incorporated three distinct contexts of use. One, as a computational agent that replaces human authors in the production of video game content, a **PCG designer**. Two, as a tool to used by human creators during video game production, with procedurally generated content being incorporated into a video game, a **PCG design tool**. And three, as part of a finished video game artefact that features in-game content generation, a **PCG game**, designed by humans with or without aid of **PCG design tools**, as in commercial games, or research-developed prototypes. In this chapter we survey PCG methods, organizing them in terms of these three basic types. Of the three contexts, what we are interested in particularly is both **PCG design tools** and PCG methods that seek to optimize player experience (as these are related to our research problem), so some case studies of this type will have further emphasis and detail.

Procedural Content Generator as a Designer

Though not always overtly introduced as such, *"the goal of procedural content generation (PCG) is often to mimic a human author as well as possible (...so as) to replace or augment human-authored content as seamlessly as possible"*(Smith et al., 2012). So it seems a logic step that similar systems may aim to replace human creators altogether, through the design and implementation of systems that act autonomously as a **PCG designer**, capable of creating content, or complete prototypes, autonomously, during production time; a process of 'Automated Game Design'.

An Experiment in Automated Game Design

While most PCG approaches focus on creating solely specific types of content, or assembling existing content, Togelius and Schmidhuber (2009) aimed at creating a system that started with nothing and ended with a complete game. Their proof-of-concept system takes a set axioms that establish the search-space of game-rules that an evolutionary algorithm will then explore. Axioms go from the spatial definition (a discrete grid 15x15) to basic rules (every game is won after a set of time, and when a score >= scoremax condition is met, the game is won; otherwise, it is lost), and define a search-space where Pac-man-like game could exist. The search-algorithm then looks for values for rule-parameters such as the time-limit, maximum score for winning the game, what pre-coded rules trigger when *things* (basically, game entities) move and collide, and so forth.

For evaluating generated rule-sets (or game designs), the system uses neural-net based simulated agent playthroughs; each agent is optimized with an evolutionary learning algorithm. Once evolved, rule-set fitness function considers several artificial agents' capability to learn how to play, giving low-fitness to games where a non-evolved agent can win, or that the best agent cannot win, and high fitness to games that can be learnt quickly.

Automated Game Design

Nelson and Mateas (2007) created a system for automated game design that designs micro-games styled in the likeness of Nintendo's WarioWare series. The system takes a noun and a verb – such as 'pheasant' and 'shoot' – and through a constraint-based system, generates variations on a small set of previously created abstract games, and then evaluating *"them according to a heuristic measure of how reasonably close they are to the player's actual request"* (Nelson and Mateas, 2007).

The process works by comparing the verb chosen by the user with prototype verbs associated with each of the abstract games; the shorter the semantic distance between the terms, the better. A set of usable nouns are then available for mapping the selected abstract game, and the closer they are to the noun forwarded by the player, the better. These nouns will serve as the basis for the selection of the sprites used in the game. With this method, the system can provide variations of simple games that thematically relate with the provided linguistic pair. As an example of the system's output, see figure 2.12.

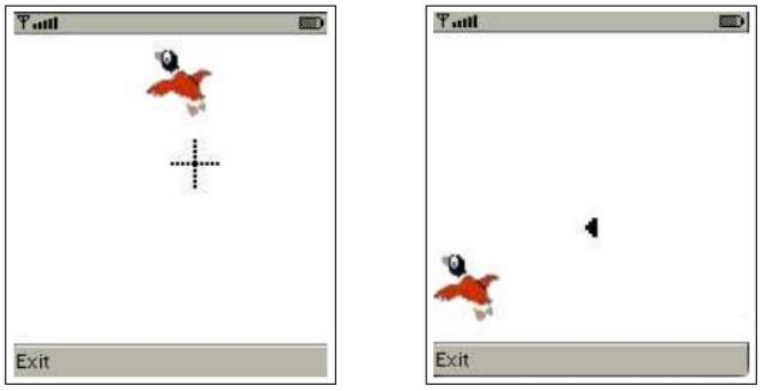

FIGURE 2.12: Example of the end results of the Automated Game Design system by Nelson and Mateas (2007):*"Two games generated for the noun 'pheasant' and verb 'shoot'. A duck is in both since it is the closest noun to 'pheasant' for which the generator has a sprite. [...] The left game is implemented by the Shooter concrete mechanic and has the player trying to shoot the duck; the right game is implemented by the Dodger concrete mechanic and has the player, as the duck, trying to avoid bullets"*.

ANGELINA

An evolved variation of the latter approach is ANGELINA (Cook and Colton, 2014), a *"cooperative coevolutionary system for automating the process of videogame design"*. It is capable of autonomously design complete video games, and all it needs from a human-being is a keyword or phrase (in the latter case it is transformed into a keyword) it will employ as a base theme for the game.

In a pre-design phase it seeks out relevant words related to the theme in word association databases, and based on each association, it retrieves multimedia assets named or tagged (by humans on Twitter) with those words. ANGELINA then generates thematic zones based on associations; each zone *"a collection of a floor texture, a wall texture, a 3D model for use as scenery, and a sound effect"*(Cook and Colton, 2014). A title for the game is established by use of rhyming dictionary and a corpus of popular culture references. Finally, a musical composition is selected, by taking the theme, finding what emotional moods are associated with it, and discovering the musical piece in a database that better fits said mood.

ANGELINA (Cook and Colton, 2014) generates designs based on four distinct co-evolutionary algorithms. Each seeks out to evolve a specific component of the ensuing video game, namely: Level Design (spatial layout of the world), Zoning (a zone map is evolved, with zones attributed to specific world locations), Placement (player entry point and exit, as well as game entities positions), and

FIGURE 2.13: Screenshots of Hit The Bulls-Spy, designed by ANGELINA-5. The top shows the game world a from a bird's eye point of view and the bottom shows a screenshot from the running game (Cook and Colton, 2014).

Rule-set (a selection of pre-baked entity behaviours, i.e, actions entities can trigger). Each system evaluates solutions based on a specific fitness function; these are either based on intrinsic qualities of the solution or data extracted from a simulated play-through of the level.

Besides the game, ANGELINA provides developer commentary describing some design decisions, using template paragraphs filled with data from the production and Internet resources. As a form of validation, 2 games designed by ANGELINA were submitted into a global game-jam, and it was evaluated by human players as outperforming, in several categories, *"many hundreds of other entries to the contest"*(Cook and Colton, 2014), even though it was nowhere near top entries.

Discussion

Though promising, these examples do not seem to warrant further research in terms of our positioning of improving game-design and production contexts. It is highly unlikely that game designers would want to relinquish so much authorial control as to let a computer do the vast majority of the design work. The quality of the solutions these systems depends on an extremely convoluted and complex generation process which game-designers could hardly be expected to control or manipulate, let alone improve.

As a side-note, their only way of increasing the likelihood of these systems delivering high-quality content would be in providing better base aesthetic and mechanical assets (which, according to Colton and Wiggins (2012), can have high impact on the end-result). Thus, as they exist currently, working autonomously, they seem capable enough to deliver working video games, in the best case, on par with amateur efforts, but hardly the sort of high-quality experience most professional game designers wish to create.

Togelius and Schmidhuber (2009) argue in favour of these techniques, mentioning that it is common in reviews to complaint that games lack imagination, and that evolutionary algorithms excel at finding unexpected solutions to problems. While we concede that the heuristic and random nature of these algorithms can help find diverging solutions for creative problems, we assess that these systems are nowhere near being able of autonomously materializing any semblance of high-quality artefact, making their capabilities for a divergent-heavy creative process effectively meaningless.

What this does tells us, is that perhaps we can use these techniques in a way that can complement human debilities, instead of attempting to replace them altogether. If one could only mesh the two – the human and the mechanical – in a way that takes the best both can offer, and avoids their pitfalls, then one could find a way of employing PCG for better design solutions.

Procedural Content Generation in Games

There are multiple cases of commercial video games incorporated with PCG algorithms that run after release, when players can play it. The most frequent use is the generation of spatial content: levels or game-world parts. The aforementioned "Elite" (Brabben and Bell, 1984) uses PCG to generate planet systems (including both spatial and game-play related variables), and in "Minecraft"

(Persson, 2011), the world map is continuously being generated (Wikipedia, 2016a). In both these examples, the justification was of a technological nature.

Other games, such as "Civilization IV" (Meier, 2005) and "Dwarf Fortress" (Adams, 2006), allow players to customize their game scenarios, providing parameter selection encompassing both spatial and game-play variables.

Also frequent is the use of PCG to pad a game's content, increasing longevity. In both "Fallout 4" (Howard, 2015) and "Skyrim" (Howard, 2011), a procedural quest generation system takes a base quest template (for example: assassin character X) and fills in details, based on player actions:

> *"where it happens, under what conditions does it take place, who wants someone assassinated, and who they want assassinated. All this can be generated based on where the character is, who he's met. They can conditionalize that someone who you've done a quest for before wants someone assassinated, and the target could be someone with whom you've spent a lot of time before."*(Bertz, 2011)

These quests, which are not part of the main or secondary quest-lines, apparently serve only to complement and increase players' play-time.

"Borderlands" (Armstrong, 2009), a first-person shooter RPG hybrid, has randomly generated weapons (Togelius et al., 2011). At any given moment of the game, and based on players' current levels, a set of basic weapon qualities are randomly attributed – element tupe, base-damage, critical chance. Similarly, in "Diablo" there is procedural generation of items and weapons and enemies (Brevik and Schaefer, 1996). "Rogue" (Toy and Wichman, 1984) and rogue-like games, such as "Spelunky" (Yu, 2014), "Binding of Isaac Rebirth" (McMillen, 2014) and "Rogue Legacy" (Lee, 2015), and to a lesser degree "Diablo" (Brevik and Schaefer, 1996) use PCG for level generation in a way that promotes not only longevity and re-playability, but also mediates Smith's Aesthetics of Challenge, Discovery and Fellowship (Smith, 2014).

However, in our assessment, generated content can also wear out as time goes by, with the quasi-similar patterns repeated after a certain variable number of play sessions. It is surely true that the search landscape of many of these games is extremely vast and potentially infinite, but minute differences are subjectively appreciated as being as redundant as no differences at all. In all these cases, the end result is a series of what we consider generic and uninviting game-elements that repeat patterns better explored in the human-authored quests.

On a purely decorative or audiovisual level, "LSD"(Sato, 1998) uses random combinatorial selection of a database of aesthetic assets (textures and sound beats), in order for the aesthetics of the game to continuously change, reflecting its algorithms' interpretation of the affective states that player actions betray, and in the process providing a trance or dream-like quality to the experience (LSDWiki, 2016).

"Blade Runner"(Castle, 1997), a video game based on the film of the same name, randomly selects several key variables before game-play starts (specifically, which characters are replicants or human), and then based on player actions (the affective inclination revealed in game-play actions and dialogue-choices), decides how specific story scenarios play out (a character might attack you out of the blue, for instance). This case is particularly interesting, in the sense that the PCG-like algorithm serves not only to improve longevity and re-playability of the game, but also to express an analogous to Deckards' existentialist questioning of his own and others nature. Human or replicant? – is the question the game thus proposes. By making a game-system in which any character – including the Blade Runner which players control – can be human or replicant, perfectly encapsulates the film's subject matter.

Galactic Arms Race

Outside of commercial efforts, research into PCG has led to the design of several PCG-games. One important reference in this space is "Galactic Arms Race", a research developed EDPCG-game *"designed to look and feel like a near commercial quality video game to effectively demonstrate the potential of automatic content generation in mainstream games"* (Hastings and Stanley, 2010). It is a space shooter with procedurally evolved weapons. These are evolved both a mechanically and aesthetically, sporting different visual effects, projectile trajectories and speed, etc.

At any moment in the game, weapons are spawned normally. As players collect and play with them, these are evolved according to a neuro-evolutionary algorithm. All weapons are given a fitness value proportional to how often players chose to play them. The higher the number of uses of a given weapon, the greater likelihood that it will be one of the parents in the evolutionary process, and therefore that its qualities will be propagated in a new round of weapon spawning. Besides this element of player influence, designers also have a measure of systemic control. They can select initial content pool (that can be evolved before game-release), and they can determine the probability of each

designed element being chosen from a spawning pool, as opposed to the natural selection process based on player-choice; this is intended to keep diversity in the content population.

To test the system in terms of creativity, 10 players played the game for at least one hour. Game-play sessions allowed generation of *"a variety of genuinely unique weapons with compelling tactical implications and aesthetics"*(Hastings and Stanley, 2010). Eventually, a version of "Galactic Arms Race" was released commercially on steam, though with less than stellar reception. In user reviews of the game's Steam page can be found numerous anecdotal evidence of its PCG systems' pitfalls:

- *"There are just a few different subtypes of weapons you can get, of which one in a hundred are plausibly usable. The hands down best weapon types for everything still are the starting weapons, or weapons with even tighter grouping of bullets than that."*
- *"i found both the previous and this version of the game pretty entertaining, although very repetitive."*
- *"The fact that weapons are randomly generated isn't as exciting as it seems."*
- *"Not really much going on in this game. Very repetitive and never felt like I was really doing much. Fly around shoot some enemies, destroy their base, and collect a weapon that looks exactly like the one you just got from the last base."*
- *"actually you only rarely change weapons once you get something you like, since most modes are pretty situational and eveolving your main weapon is actually risky"*
- *"Galactic Arms Race is built around a really cool core idea. Weapons, thousands (millions? trillions? bazillions?) of them are procedurally generated and evolve based on your preferences. (...) The weapons, which are the sole focus of the game, are at the same time impressively varied and boringly repetitious. The variety of projectile patterns is astonishing [... but] for all the variety in their movements and colors, in almost every other way, every weapon in the universe is identical."*

While the aggregate of all reviews is summed up as 'Neutral', there is a non-negligent number of negative reviews where, apart from finding criticisms to the game-play, users question the very nature of the PCG system. We think this is a symbolic example of the classic problem with PCG algorithms: greater quantity of generated elements does not abate the sense of repetition of the same patterns being exposed over and over again. The quest for added re-playability in these systems exposes their limitations, and is not in any way a guarantee of

an improved game experience... only a longer one.

Facade

"Facade" (Mateas and Stern, 2003) is a short, research-developed, interactive narrative game, where players have an *"experience akin to being on stage with two live actors who are motivated to make a dramatic situation happen"*(Mateas and Stern, 2003). The synopsis reads,

> *"you (...) play the character of a longtime friend of Grace and Trip, an attractive and materially successful couple in their early thirties. During an evening gettogether at their apartment that quickly turns ugly, you become entangled in the high-conflict dissolution of Grace and Trip's marriage. No one is safe as the accusations fly, sides are taken and irreversible decisions are forced to be made. By the end of this intense one act play you will have changed the course of Grace and Trip's lives (...)"*(Mateas and Stern, 2003)

Both Grace and Trip move, talk, gesticulate, etc, and the player, through a natural language input window and a mouse pointer, can reply to them in kind. An AI system, the drama manager, controls story flow by selecting story beats, the *"smallest unit of dramatic action that moves a story forward (...) annotated (...) with preconditions and effects on the story state, instructing the drama manager when they make sense to use, in the interest of creating an overall dramatic narrative ?– a plot"*(Mateas and Stern, 2003).

The generative quality of "Facade" comes precisely by the way this drama manager *"mixes and sequences behaviors in sophisticated ways"*(Mateas and Stern, 2003). Meanwhile characters AI agents *"attempt to engage the player in psychological head games, for example, posing situations to the player in an attempt to force her to choose sides in an argument"*(Mateas and Stern, 2003). Player actions then affect his/hers affinity with both sides of the couple, tension level, and information that is revealed about their marriage.

The System's architecture uses a reactive planning language called A Behavior Language (ABL), used for modelling Grace and Trip and the Players' agent; a Natural Language Understanding template language to process player's textual input; a Reaction Decider language to decide replies to player discourse, and a Beat Sequencing language for the Drama Manager that specializes in deciding which story beat gets selected.

The game strives for an interesting middle ground between structured narratives and simulations; like the former, it provides a unified, efficient experience

with fluctuating tension to resemble an Aristotelian dramatic arc, and like the latter, it permits greater player agency and freedom of expression. The system still requires a consuming authorial process (Mateas and Stern, 2003), as all story content is hand-created at the atomic level by humans, including the definition of agents' behaviour, and story-beats' constraints and effects.

"Facade" is surely not perfect: players' immense freedom might provide agency, but comes at the cost of allowing several awkward or simply absurd interactions with the agents. Despite this, its rather unique combination of player agency and authorial control is very promising (and to this day, as far as we know, unbeaten). The core philosophy behind it is simple: provide mechanical freedom to players, and then steer the system's actions so as to orchestrate a cohesive, structured narrative experience.

The computational aspect of "Facade" lies only in the interactive and dynamic feeding of painstakingly created, high-grade human content; there is no focus on on-the-fly generation just for the sake of endless playability. While it is a fact that "Facade" was designed to be re-playable, it can work when experienced in a single play-through – as the system's reactions come across as dynamic and unique and believable –, and we argue, is all the better the less one over-exploits its 'generation' system.

Furthermore, while "Facade" exists as a single materialised artefact made possible with its system, a PCG-game, it was built as a proof-of-concept for a tool intended for human authoring, as there is no generation proper at work, only a highly dynamic selection of human-authored content. This focus on the human factor also helps explains why it is of the highest quality (though its development came at a significant cost (Mateas and Stern, 2003)).

There was never a commercial game that applied "Facade's" system, but there was one with a seemingly similar philosophy. "Left4Dead" (Booth, 2008) and its sequel (Booth, 2009), are collaborative multi-player shooters where a director AI reads player actions, and then dynamically controls game events – namely, enemy and item placement, spawn timing, character communication, level music, map configuration and visual mood selection – so as to modulate player tension, stress, pacing, conflict and difficulty in a way that resembles a dramatic arc (WikiA, 2016). Again we find that, besides increasing re-playability and longevity of the game (given the possible situational variation the system provides), there is in here an expressive use of PCG systems.

Like "Facade", the system basically adapts the game-play flow and aesthetics

so as to optimize the experienced dramatic structure – slow-paced first (low-level enemies, abundant resources), rising tension in the middle (growing enemy count and difficulty, diminishing resources), tension peak near the end (attacks of waves and waves of enemies and boss-like creatures in a short period of time), and a cathartic denouement in the end (shorter, but similar to the beginning).

For the effect to work, some sort of player model must be implicitly encoded in the Director's reading of players' behaviour (so as to find the right amount of antagonism to throw at players without breaking the experience), though we could find no exact record of what data is used and how it is processed (supposedly, it processes players' status, skill, and location (WikiA, 2016)).

In these examples, we find, even if only in a tenuous form, an alternative way of using PCG: as a way to guarantee that, in a given play-session, players go through certain patterns of in-game actions that, implicitly, realize the experiential agenda intended by authors. More so, we consider "Facade" an interesting video game, and both "Left4Dead" games were commercial and critical successes (*Left4Dead*; *Left4Dead2*), which points, at the very least, to this approach's potential... unlike other examples.

Endless Web

"Endless Web" (Smith et al., 2012) is a platform game that uses PCG algorithms to enable new kinds of game-play. They define it as a PCG-based game *"one in which the underlying PCG system is so inextricably tied to the mechanics of the game, and has so greatly influenced the aesthetics of the game, that the dynamics player strategies and emergent behavior revolve around it"*. Besides the system generating level geometry indefinitely, players can actively and consciously increase frequency and difficulty of level enemies.

Additionally, aesthetic assets (both tile art and music) are procedurally selected, based on player location and previous player actions (for example, the more difficult the current setting, the more chaotic the selection of music). 'Endless Web" also controls rhythm and pacing parameters of the level generator, increasing them as players reach goals, upping the tempo and, consequently, the difficulty of platforming.

The authors conclude their work with a series of guidelines referring to challenges faced during the design of their project. First, balancing control over content:

> *"Control over the content generator must be balanced between the player and the game itself. When creating a PCG-based game, the role of the game designer shifts from being a creator of instances of content to an entire, parameterizable range of content. The challenge here comes in ensuring that, while the content the player experiences will be varied, the overall game experience is still appropriately controlled."*(Smith et al., 2012)

Second, keeping the PCG visible to the player, basically meaning that if players do not understand how the world is being generated around them, they can lose agency. And third, building support for art, i.e. *"design the system to be flexible enough to accommodate a changing art or music style over the course of the game's design"*(Smith et al., 2012).

Adapting Playgrounds for Children's Play using Ambient Playware

Experiments in adaptive game experiences are still few in number and are mostly small steps that pale in comparison with the promising concept. One early experiment is that of Derakhshan, Hammer, and Lund (2006). The goal was creating a playground game that while capturing the appeal of other entertainment forms (such as video games), could maximize physical activity of children. The solution is based on playware – a computational playground composed of a floor-like device built out of squared tiles, each with a pressure button, a light capable of emitting in 4 different colours and a communication system. Each playware tile can measure 3 interaction variables: pressure applied to the tile, distance travelled in one movement (between tiles) and duration of the movement. The authors defined two states for these variables: soft and hard, near an far, fast and slow, respectively.

A very simple game called bug smasher – essentially, a game where players must 'smash' tiles coloured red – was implemented. For the system to work, besides the interaction variables, the Playware environment provided a tile-state variable (pertaining to the game's state, with or without a bug), a neighbourhood variable (states of the other tiles) and finally, direction of movement. The authors then created a neural network that could take these and classify them into 8 different classes; their rationale and description is shown in figure 2.14.

A multi-layered perceptron was trained with back-propagation algorithm, with training data obtained via subject-led video analysis of a play session. Classification results rounded 96% for 5 of the classes (3 were never observed), which means that 24 out of 25 children behaviours were correctly classified.

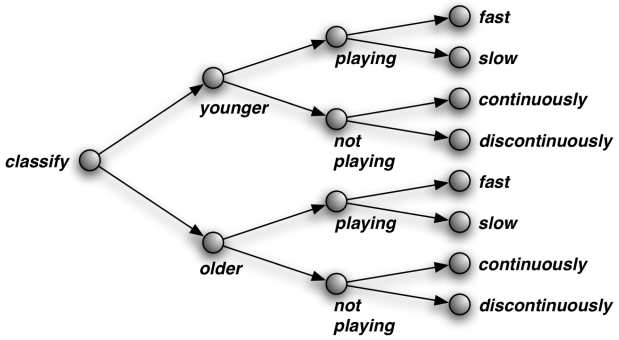

FIGURE 2.14: Playware's 8 categories for children's behaviour from (Derakhshan, Hammer, and Lund, 2006).

As the neural network proved extremely accurate, the second phase of the project involved using it to adapt, in real-time, the game-procedures. Bug speed was regulated proportionally in reference to player speed, distance of the bug to the player is increased for older children and vice-versa, and if the child is not playing, the bugs start a follow behaviour, placing themselves as close as possible so as to lure them into actually playing. These adaptations were intended on maximizing children's motivation to engage in physical activity. The validation of the adaptive version was made with children playing, in 90 second periods, the static followed by the adaptive version (order effects were assumed non-existent). Results showed statistically significant variations in several key physical activity measurements, such as the total number of interactions, average distance, average force, etc.

These results indicate that the system was capable of stimulating subjects activity more effectively than the static version. While a very basic form of player experience, this simple solution paved the path for more complex forms of optimization of player behaviour and experience.

Optimizing Player Entertainment

Yannakakis and Hallam (2004) studied how to measure game interest in predator/prey games (such as Pac-Man), based on the assumption that interest is mostly determined by qualities of computer characters' behaviour. To this effect, they proposed a neuro-evolution learning algorithm intended on maximizing interest criteria. These criteria were based on previously researched models of what constitutes interest, and then formalized into a set of measurable metrics:

– Difficulty of the game – neither hard nor easy. Measured in the time it takes to kill Pac-Man in each game. Maximizing values that distance themselves from the optimal approach and the worst possible strategy.

- Behaviour diversity – ghosts should use different ways of killing Pac-Man, to make them seem less predictable. This metric is measured in terms of variance of the time it takes to kill Pac-Man, maximizing the diversity of time in killing Pac-Man.
- Aggressiveness in behaviour – ghosts should be simultaneously trying to kill the player while also exploring as much of the game field, to give an illusion of strategic planning. This is measured in terms of movement entropy on part of AI movement, quantifying the completeness and uniformity with which ghosts cover the field.

The criteria formally defined, the experiment carried out evolving of ghost controllers that maximized an interest metrics (that is calculated as a linear product of the three criteria) while playing against simulated Pac-man players with different behaviours. The AI controllers were tested in both off-line and on-line modes, showing they was capable of achieving high interest results independent of Pac-Man play styles, map types and even being capable of on-line adaptation to variation of said styles (Yannakakis and Hallam, 2004; Yannakakis and Hallam, 2005).

Following the study, Yannakakis and Hallam (2007) validated their interest metric by correlating with self reported interest values. They tested their on-line adaptive AIs against a significant number of subjects in a battery of tests. In them, each subject was called to play a pair of games against the AI opponents and then fill out a 2-alternative forced choice questionnaire, by choosing which game was more interesting – the first, the second or neither. Results from the test series showed high correlation between players' interest ranking and the one derived from the computed interest metric. While this was not verified in all test instances, results were statistically significant, allowing the authors to confirm their hypothesis that the interest metric was consistent with human players' subjective judgement(Yannakakis and Hallam, 2007).

In this case, results show that not only player behaviour can be optimized, but specifically form of player behaviour that has high correlation with player's subjective judgement. This form of optimization, though focused only on one experience quality, shows that EDPCG methodology can be used to fine tune a game so that it is guaranteed of mediating a given experience blue-print.

Modelling and Optimizing Player Experience

Pedersen, Togelius, and Yannakakis (2009) used metrics data from several game-play sessions to track correlations with player experience reports. 16

different Super Mario Bros. levels were generated by varying between 2 different values (high, low) for each of the following control variables: number of gaps, average gap width, spatial diversity of gaps and number of direction switches. Other design features were maintained as static. Tracked metrics included variables such as (but not limited to) – number of jumps, level completion time, collected items, number of deaths, number of enemy kills, etc.

120 subjects were invited to play the different levels, playing 2 different variants twice, one in each order (variant A then B followed by B then A or vice-versa). After each player experienced two games twice, he was asked to fill out 4-alternative forced choice questionnaire, comparing them in terms of relative emotional impact, referring to the emotional categories of fun, challenging and frustrating.

Statistically significant correlations were found between several metrics and the emotional categories. Amongst the major results of this analysis was found a common thread of positive correlation between variables that signify unhindered progress with fun, and variables that symbolize difficulty and loss with both challenge and frustration. Fun was not correlated to any of the featured game design variables, though frustration and challenge were. Significant correlations are found between the three emotions, with fun and frustration naturally negative correlated, but somewhat surprisingly, both fun and challenge *and* challenge and frustration were found positively correlated. This results hints at a non-linear relationship between these emotional categories.

For further exploration of these relationships, the authors trained a simple, one-layer neural network paired with a feature selection method, to try and predict emotional values based on the metrics. Results showed that challenge and frustration could be predicted high levels of accuracy (77.77% and 88.66% respectively), though fun could only be predicted with less powerful results (69.18%). These results underline the non-linear nature of the relationship between these emotional categories, but also the difficulty in assessing fun based on objective data. Perhaps this is a sign that fun is too a subjective or too a complex term to be predicted across different subjects. Nonetheless, the most interesting results are the correct classification results associated with the other two emotions. This hints at the possibility that, at least these two emotional categories are prone to prediction based solely on objective, measurable data from play sessions. If this hypothesis can be confirmed and further explored, it makes it possible that an emotional predictor can be placed inside games

themselves, allowing the game to infer the emotional state of its players. Should this be the case, several different avenues are opened for experience evaluation.

This study was then expanded to cover 6 emotional categories: fun, challenging, boring, frustrating, predictable, and anxious, following a similar methodological approach as before (Pedersen, Togelius, and Yannakakis, 2010). Fun was once again found to be correlated with an easy, consistently progressing experience, though correlations were not as strong as with other variables. Conversely, challenge was related with increased difficulty. Frustration associated with idleness (according to the authors, symbolizing increased time in planning the next steps in the game, on account of difficulty), lack in obtaining positive rewards and losing. Boredom was only correlated, with statistical significance, with gap width. Predictability was correlated with gap width, frequency and difficulty, and was negatively correlated with finishing the level. Anxiety was found to be inversely correlated with the same variables as predictability. Predictability and Anxiety are therefore negatively inter-correlated; Challenge and Anxiety are positively correlated and statistically significant.

Finally, their model's accuracy for the six emotion types was: fun – 74.21%, challenge – 79.37%, frustration – 91.33%, predictability – 76.28%, anxiety – 77.78% and boredom – 73.19%. While still far from ideal, these results are groundbreaking in that they show, once again, strong emotion prediction capability, in terms of breadth and accuracy, based only on level design data and game-play metrics.

To conclude, and using the same data collection strategy and with improved search methods, the 'fun' emotion was optimized for both real and simulated players, using on-line generation of new levels (Shaker, Yannakakis, and Togelius, 2010). Results show statistical significance and some generalization power, thus presenting a convincing case for future use of this approach in a commercial context, so as to improve entertainment value of produced levels or inclusion of automatic level generators that can cater specifically to each player. Furthermore, the wider spectre of emotions that were tested show that EDPCG can be expanded beyond unidimensional metrics such as fun or interest, and made to incorporate a varied range of emotions and cognitive judgements of the experience.

Procedural Content Generation in Design Tools

Certain areas of video game content are frequently created by resourcing to Procedural Content Generation in the design phase. Landscape generator

Terragen (Software, 2016) is usually employed in film, TV commercials and photography, as a way of producing low-cost backgrounds and pictures, but it has also been used at least once in a videogame, "Serious Sam 3", to create its skyboxes (Wikipedia, 2016b). More frequently used in this medium is "Speedtree", a foliage generator that has been outsourced in several commercial titles (IDV, 2016), the most prominent of which the recent "Witcher 3"(Tomaszkiewicz, Kanik, and Stepien, 2015). Similarly, there is "Grome", a terrain generator that can be used to generate game-levels or game-worlds (QuadSoftware, 2016).

Despite the immense usefulness of PCG and EDPCG methods, surveyed state of the art solutions did not research, with any form of meaningful detail, how they could be introduced into existing game design and production contexts of use; they were created to semi-autonomously solve a very specific design problem. It is surely true that the goals of engagement, interest and fun optimization are pursued by many game designers and production companies. Yet video games are a class of artefacts in a creative medium, and the idea that game design can be reduced to the process of optimization of a single experiential variable – whichever it may be – is a simplification. Granted, it is a useful one for research purposes, but one will be difficult to extend beyond the research laboratory.

How can EDPCG methods be appropriated in design and production contexts, operated by human designers, and made to work for them instead of simply replacing parts of their work? This is the question which other research efforts seek to answer. The key difference in the following case studies, is their shift in perspective away from the inner workings of the computational algorithm taken in isolation, to its workings when paired with a human designer. Instead of contributing with computational models and processes that can successfully solve a game design problem, these efforts seek to find algorithms that, in the hands of a designer, can help solve a game-design problem.

Tanagra

Smith, Whitehead, and Mateas (2010a) share our preoccupation with the fact that *"existing techniques in procedural level generation tend to offer little in terms of author guidance"*, aside from parametrization that can lead to unpredictable results. They hold a *"mixed-initiative approach to design, where content is created through iterative cycles between the human designer and a procedural content generator, capitalizes on the strengths of both human and computer designers"* (Smith, Whitehead, and Mateas, 2010a).

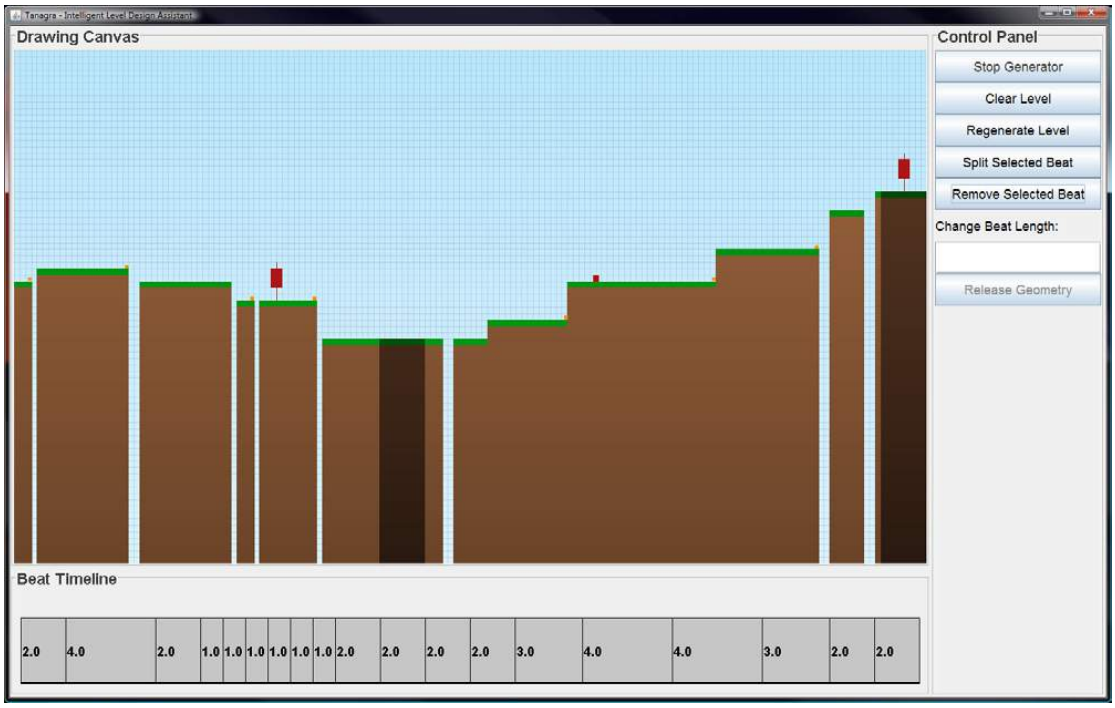

FIGURE 2.15: Tanagra's (Smith, Whitehead, and Mateas, 2010b) interface. The bottom of figure shows the beat timeline and the top the actual level that is being edited by the human designer and generated by the system.

Tanagra (Smith, Whitehead, and Mateas, 2010a) works similarly to a platform-game level editor with two layers: tiles (basic geometric elements of the level: platforms, gaps, springs, enemies, and stompers) and beats (building blocks of rhythms, they represent single or multi-step player actions such as jumping over pits, jumping to kill, running and then stomping, etc). Besides directly adding and editing tiles, designers can add, remove and modify (either resizing or splitting) beats; order the system to generate new geography for either the whole level or a single beat; and 'pin' and 'un-pin' tiles so as to determine which tiles Tanagra can manipulate or not. Tanagra then generates, either when warranted by users, or autonomously, level variations in accord with user-created tiles and beats, producing a new iteration in a matter of milliseconds. To the authors, this is what allows designers to control both the *"physical properties of the world"* by way of editing tiles directly, and *"properties related to potential player behavior by influencing the pacing of the level"* (Smith, Whitehead, and Mateas, 2010a), by way of beat editing and consequent procedural generation.

Generation occurs by way of a Reactive Planning Language, ABL, that determines, from a set of geography templates, which should be forwarded to instantiate a given beat. There is no explicit qualitative evaluation of levels;

instead, Tanagra has Choco, a constraint solving library with a set of encoded level constraints (e.g. the exit point of one beat is on the same level with the point of the next beat so that they line-up) that it uses to validate generated content. This, in turn, guarantees that each level is playable. Tanagra is capable of producing a large quantity of level variations, and its generation capabilities were evaluated in terms of both non-linearity and player leniency (a ratio the number of player-opposing level features) of produced levels, revealing a slight bias towards more linear, less lenient levels.

Tanagra is a rather interesting compromise between using PCG to facilitate level design while affording designer control over the end-result. Beat editing allows designers to decide the overall flow and rhythm of the experience before the computer fills in the details; and whenever designers so desire, they can hand-craft minutiae, retaining their authorial control over the end-product. It represents a massive step forward, especially when compared to other state of the art PCG systems, that do not take into account human factors.

To us, Tanagra looks particularly adept at facilitating rapid prototyping of levels, (therefore increasing production efficiency), or as a means for generating near endless variations of the same designer-made beat templates, thus serving to increase a video game's re-playability, and mediating player aesthetics listed in Smith (2014) (in fact, as we have previously reviewed, "Endless Web" (Smith et al., 2012) seems to do just that, and shares some of Tanagra's basic logic). However, as a creative tool for designers, Tanagra's benefits seem more modest to us, for despite its increased level of user parametrization – a significant contribution – it still only offers the same basic benefits of PCG: quantity in content at a sale price.

Sentient Sketchbook

Besides several contributions into EDPCG, Yannakakis, Liapis, and Alexopoulos (2014) and Liapis, Yannakakis, and Togelius (2013) went beyond the creation of methods for generating content, and researched into Mixed-Initiative Co-Creativity (MI-CC) solutions, i.e., for PCG-like methods where both human and computer pro-actively contribute to solutions. This marks a departure from EDPCG in the sense that there is an explicit formalization of the human role when interacting with the PCG system, one that goes beyond *"choosing parameters for the generation algorithm"* and *"editing the generated artifact"* (Yannakakis, Liapis, and Alexopoulos, 2014). Sentient Sketchbook allows *"human designers to edit a strategy game level while computational creators are simultaneously creating*

variations of the user's level". Designers create levels by sketching them in an editor; sketches are abstract, low-fidelity representations of levels that are quick to comprehend, evaluate, and edit, and that can at any given moment be converted to playable levels.

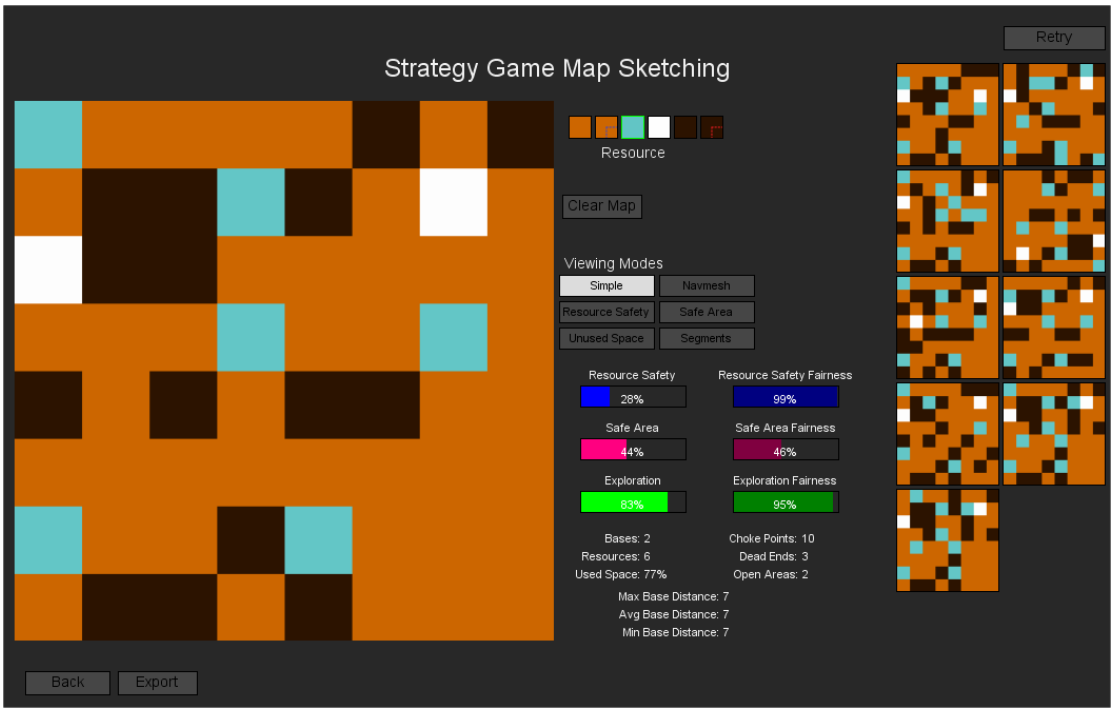

FIGURE 2.16: Sentient Sketchbooks's (Yannakakis, Liapis, and Alexopoulos, 2014) interface. On the left is the current sketch editor and on the right are the computer-generated alternatives. In the mid section sit, from top to bottom, the tile palette, alternative display map options, and a number of automatically computed evaluation heuristics.

Computers assist designers in two ways. First, they evaluate levels according to a number of domain-specific properties and heuristics; these include aspects such as resource safety, used space and safe area fairness. Also, the tool offers visual aids, such as a *"navigational grid, optimal paths between bases, or "safe" resources around each player base"*(Yannakakis, Liapis, and Alexopoulos, 2014) Second, permutations on users' level sketches are suggested by the system; these are generated via a Genetic Algorithm which besides guaranteeing their playability seeks to either *"optimize the domain-specific level evaluations used by the tool, or to create maps that are visually diverse from the user's sketch"*(Yannakakis, Liapis, and Alexopoulos, 2014).

The aim behind this tool's design is to stimulate Diagrammatic Lateral Thinking. Diagrammatic reasoning – reasoning via the use of visual representations

- is stimulated through the use of sketches (as opposed to levels themselves). The tool's generated solutions, especially those that are optimized to diverge from the users' own, operate on the basis of the random stimulus principle of lateral thinking, as it relies on *"the introduction of a foreign conceptual element with the purpose of disrupting preconceived notions and habitual patterns of thought, by forcing the user to integrate and/or exploit the foreign element in the creation of an idea or the production of a solution"*(Yannakakis, Liapis, and Alexopoulos, 2014). Diagrammatic Lateral Thinking fuses both these principles.

To evaluate the tool, a small-scale user study was carried out with 5 independent game developers, game programmers and level designers. Each participant was asked to use the tool at their own leisure. Each participant had several distinct map design sessions; data from 24 design sessions was analysed, with in each session starting from a blank canvas and finishing with a complete map. According to the research authors, expert users *'were overall positive about the simple interface, the different visualizations and the evaluated metrics and fitness dimensions which reduce user effort"*(Liapis, Yannakakis, and Togelius, 2013). However, their *"expert knowledge was sufficient for designing "good" maps, and generated suggestions were thus often overlooked"* (Liapis, Yannakakis, and Togelius, 2013). Thus, we argue, the usefulness of the tool was mitigated, at least for an expert-oriented context-use.

2.2.4 Discussion

The Issue with Game Design as Optimization

On the forefront of research that aims to operationalize experience evaluation methods and Artificial Intelligence methods has been EDPCG, with a great focus of player-experience optimization (Pedersen, Togelius, and Yannakakis, 2009; Pedersen, Togelius, and Yannakakis, 2010; Yannakakis and Hallam, 2006; Togelius, Nardi, and Lucas, 2006; Shaker, Yannakakis, and Togelius, 2010). In the reviewed EDPCG studies, the optimized experience variables were improved based on players' own perception of entertainment, or interest, or fun; as such, what systems have been optimizing, so far, is not actual entertainment/interest/fun but its perception by players. As we have previously detailed, humans are prone to misreport their feelings and cognitive appraisals in self-evaluations, for a number of well-known psychological factors (Jenkins, Brown, and Rutterford, 2009; Ekman, 1999; Dillon et al., 2000; Rottenberg, Ray, and Gross, 2007).

As such, we question if by asking subjects to assess their experience, and then designing the system to model such patterns, these experiments are really improving actual experience, or only subjects own misconception of an improved experience. And this brings us to a fundamental question regarding research in the EDPCG family. Behind its concept lies a somewhat pernicious ideal: that experience data can be used to model player's profile, and then use it to dynamically tailor the game to satiate players' desires and increase their perceived satisfaction. But is this a path worth pursuing in a design discipline?

Choosing to define experience quality in respect to user preference, while research worthy and certainly useful for certain production contexts, can not be the only path. By submitting to what 'players want' or 'need' (or think they want/need), and by using a vocabulary rooted in ill-defined concepts such as 'fun', it may be missing the whole point of a creative media such as video games. As such, in diametrical opposition to the notion of optimizing entertainment, but employing similar methods, we propose that there should be author-centric appropriations of the same techniques, not intent on optimizing player experience, but on empowering designers' agendas.

The Magic Formula

For centuries, the act of creation has been a burden rested upon the human author: whether alone or in a collective, human agency has been the driving force of the creative act. Reviewed PCG methods show this has changed, as computer algorithms are now able of creating content autonomously or, at least, semi-autonomously face human authors. And despite the human bias against computationally created artefacts, there have been several successes in Computational Creativity areas.

Research in Computational Creativity has led to the creation of autonomous generative systems in a wide array of domains: non-photorealistic art, music, jokes, stories, mathematics, engineering, among others (Liapis, Yannakakis, and Togelius, 2014). More so, the cultural value of artefacts produced by these systems as been validated by humans: *"prints of artworks by The Painting Fool have been sold; theorems from the HR discovery system have appeared in the mathematical literature; the Ludi system has invented a popular board game, for which people have paid money; and the Continuator jazz improvisation system has played alongside professional musicians"* (Colton and Wiggins, 2012). However, we ascertained several key issues with current examples of procedurally generated content, namely, how there is a tendency for repetition and lack of distinctiveness and character. The

reason is why?

Standard computer programs are not intelligent, they operate by strictly following a plan of procedures, rules, or commands – the program – that were written by a human being – the programmer. Computer algorithms that automatically create content for, or complete versions of video games are merely implementing a formula for video game content creation. In this we find the a philosophical doubt we can direct towards computational creativity approaches to solve creative problems; can a set of rules or a formula for a given type of creative problem really be found? Surely, PCG solutions we reviewed show neat and efficient approximations to solving creative problems, but we find them lacking immensely when viewed in light of state of the art hand-authored solutions (this is a steep comparison, for sure, but if one views PCG as a method for obtaining high-quality solutions, it is the only viable comparison).

Our own observation of this problem leads us to think that a part of the problem lies in the very axiomatic fabric PCG solutions sustain themselves. While not always explicitly stated as their goal main goal, PCG research may be attempting to write a formula from which high-quality content – on par with human-authored – can be derived repeatedly. Such goal assumes three things: one, that such a formula exists, two, that quality is impervious to subjects' sustained exposure, and finally, that it can be computationally translated. We have reasons to believe the first two, at the very least, are false assumptions.

First, in the history of art (Gombrich, 2009), never has such a formula been created. Even the most genial artists have only temporarily held the key to creating high-quality artistic objects, and even then used very distinct styles. Throughout history tastes changed constantly, and what once was deemed beautiful and sublime was no longer thus considered afterwards, and vice-versa. These are not flukes: our individual and societal taste patterns fluctuate regularly, and with it the perceived merits of particular artistic creations. Technology and styles also fluctuate feverishly, always in a frantic search for novel tools and techniques to create novel artefacts. Novelty is, in of itself, a feature of art. And novel artefacts, to be deemed creative, are novel in the sense they are made with new processes, new rules and new ideas of value, dynamic at heart.

Ernst Gombrich, notable art historian, famously said: *"One never finishes learning about art. There are always new things to discover. Great works of art seem to look different every time one stands before them. They seem to be as inexhaustible and unpredictable as real human beings"*, and that *"one cannot explain the existence of*

genius. It is better to enjoy it" (Gombrich, 2009) Imagine to not only explain, but replicate the essence of genius, the very formula underpinning creations of a given type or author... this seems to us absurd.

It is the most daring of projects: to create not only an artificial artistic genius, but an artificial, endlessly creative artistic genius. This is not to dismiss reviewed PCG approaches, only their underlying motivation into the creation of algorithms capable of producing high quality objects in quantity. Examples from our literature review suggest that quality and quantity are not compatible; each, in isolation, may be achievable and we have found several examples, but not both in simultaneity.

Now, this does not, in any way invalidate all the great strides made in this area, as imperfect steps to a potentially unachievable goal. These approximations are interesting in themselves, and can provide a myriad of solutions to other problems, even if not the grander goal, as the scientific and technological ground that is being broken in this area leads to many important contributions that should naturally be further procured. But, unfortunately, several PCG approaches are attempting to solve both in tandem: to not just try and create the algorithm that can, like humans do, stumble upon one or two or a great works once in a full moon, but to attempt to find the hidden formula that can consistently deliver a vast number of great works in succession. This, it seems to us, is a Chimera.

Chapter 3

Research Problem

In this chapter we define the scope of the Research Problem we aimed to solve, and the strategy adopted towards achieving that goal. In the first section is a summary of the major conclusions extracted from the Literature review, and how they helped constrain said Problem. In section 3.2, the research questions we answered are listed, followed by the chosen methodology 3.3. Finally, the research steps taken to implement it are in section 3.4.

3.1 Problem Constraints

Here we briefly sum up our state of the art analysis, by listing several key findings referring to both game design and research, which inform our research proposal, and constraint the problem we will solve.

Conclusion 1. Game design and production processes require significant effort and cost, specifically, in repeated cycles of creation, testing and evaluation of prototypes.

Our main area of interest is in improving game design processes, irrespective of the production context. The relevance of this conclusion is that it shows an opportunity for the improvement of these processes, which can have positive impact in this design practice. If one would design information systems that could partially automate these processes, while providing better information access to designers, and more powerful means for intervention, it could significantly improve the quality of the process (see section 2.1.1 for more detail on the process itself).

This is our research's purpose and intervention scope: to aid game designers in creating, evaluating and testing their game prototypes. So, the first part of our

research agenda lies in studying *what computational means can be offered to game designers that can assist them in the creation, testing and evaluation of prototypes?*

Conclusion 2. Contemporary design practices focus on mediation of experiential qualities, as opposed to crafting material qualities.

This means that any intervention in the game design process that seeks to improve it should view experience mediation as the primary goal. A better game design process empowers designers to produce artefacts (in this case, video games) that mediate a user-experience that has a set of ideal qualities. In other words: players' experience – what they feel and think, and how they behave in respect to the game – should be actively shaped by the design process until it is adequate; the game-object understood purely as a means towards that end. If we take this idea in light of the previous conclusion, we come to the notion that game prototypes should be evaluated and tested in terms of how their mediated player-experience fits a given ideal.

But what can constitute this ideal, and who should make the underlying judgement of quality? In this we make an ethical valuation of authors roles. While we have found indications that there is a bias in favour of what players want in game design research, we willingly position ourselves in the field of the authors, for we consider them to be the driving force of the creative process. Thus, they should be given free reign to determine which qualities they want their artefact to express to players, and it is our role as researchers to provide the tools that can aid them, in their goal... whatever it may be.

The change of focus from players to designers is based on the idea that art has, for the most part, been sustained by authorial pursuit and vision, despite not always having been well accepted by audiences or critics in certain historical periods (Gombrich, 2009). We consider that putting too much emphasis on their judgement may severely hamper novelty in the creative process. Given that historically video games have been criticized for their lack of creativity (Costikyan, 2005), we think we should avoid incorporating any bias that can (potentially) hamper it.

Even in non-artistic production contexts, authors should have extensive knowledge on what constitutes a powerful experience, based on previous study and work portfolio, and this provides them with the linguistic, cultural and analytical background that users often do not possess. This makes their input that much valuable, profound and well supported, while most users opinions',

by lack of formal training and education, may be superficial and dismissive (Gladwell, 2005). Which is why we favour empowering the designer to express his/her vision, even should it occasionally fail to resonate with players (whether or not the design is good or bad being besides the point). Authors should determine which qualities they want their artefact to express to players. The corollary to this stance is that any tool we afford designers should offer them as much flexibility and plasticity as possible.

Note we are not oblivious to the relevance of player-consumers in the market, or that of designers that strive to fulfil their needs. We merely maintain an ethical imperative to not discriminate in favour of that sub-set of design projects. Using an analogy, we want to provide the means for painters to paint, not to provide the means for painters to paint (what a given audience considers) pretty pictures. This is, we believe, the only way to foster creativity, experimentation and innovation.

Consequently, we added a clause to our research that corresponds to this stance. Thus, we aim to search for *solutions that can aid game designers in the creation of prototypes that mediate their intended player experience*.

Conclusion 3. Game research has not yet delivered sufficiently robust models for mapping the relationship between game design elements and their mediated player experience qualities.

As seen in section 2.1.4, existing models for relating player experience with material game elements are still imperfect; i.e. they do not allow designers to accurately predict how a specific game will affect players' experience. Like Conclusion 1, this highlights an opportunity for action. Because designers cannot predict how player experience qualities will emerge from contact with the video game object, they navigate the design process based on imperfect knowledge. Therefore, any breakthrough that can help in this process should be useful to them.

Conclusion 4. There is an abundance of methods for evaluating user experience in the video game medium. Game-play metrics, among others, require a significant effort to operationalize, in part due to challenges in interpretation.

Experience evaluation methods can assist designers' process by affording them data on player-experience. However, they come with caveats which can

negatively impact their utility to designers (see section 2.1.3 for details). Gameplay metrics – the most recent of these methods and one of the most powerful – offers highly detailed data on player-behaviour, but requires considerable effort to interpret. More so, even when correctly interpreted (meaning, resulting in a correct evaluation of the player-experience mediated by a game design), the question still remains of how to then operationalize such an assessment into a new game design that improves on the previous iteration. All these missing links in the cyclical chain – from designed game, to experience evaluation, to re-design and so forth – signify an opportunity for improvement and automation.

Conclusion 5. Procedural content generation offer methods that, without direct designer intervention, can evolve game content with a view to maximize (certain) experience metrics, by using state of the art artificial intelligence algorithms.

If one could devise methods that facilitate the process of collecting and operationalizing experience data, for example, by creating tools that use it as a basis for automatic or semi-automatic refinement of game artefacts, then we would be assisting game designers in the process. As we have seen in the previous chapter, several different Artificial Intelligence-inspired procedural content generation approaches (section 2.2) have been proposed that automatically evaluate and design video games. As such, these might me a method for improving the game design process.

What the previous two conclusions show, is that the possibility of creating a tool capable of analysing players experience and then modelling it in a video game in order to realize certain experience goals, without the need for human interference, is within reach. We argue that the question then becomes not only one of technological feasibility, but of how to contextualize this new power in order to make it usable to game designers. In other words, how to frame this new technology in interfaces that serve authors, empowering their creative expression? So our main research question will be *How can game designers use advanced experience evaluation methods and procedural content generation algorithms to prototype video games that mediate intended forms of player experience?*

Conclusion 6. There is a trend in Procedural Content Generation approaches to attempt to replace human authors, improve entertainment value, add replayability and player customization to generated content. Authorial considerations are either absent or, at best, secondary.

As of now, certain Procedural Content Generation approaches already solve

similar problems to the one we just stated. Yet, from our point of view, there is a significant issue with how Procedural Content Generation techniques have been positioned in terms of impact in the game design process. Procedural Content Generation approaches have been seemingly influenced by an Artificial Intelligence research mindset, replacing human factors in the design process, or at the very least, removing them from the creative helm.

In light of previous conclusions, we would like to appropriate these novel computational methods as means to support game designers in their practice. We wish to position this work in line with artefacts such as Tanagra (Smith, Whitehead, and Mateas, 2011) and Sentient Sketchbook (Yannakakis, Liapis, and Alexopoulos, 2014), where PCG is used as a tool to assist designers, and not to replace them. Rather, our aim is to give authors a set of tools that can help them actively influence what the player-experience should be like.

Also, it must be noted that in cases of optimization approaches (Yannakakis and Togelius, 2011), there is scant research showing how game designers would interact with these systems so as to improve the game design process. Methods are proposed *as is*, without studying or explaining how they would be used by authors.

We do not seek to propose novel methods adept at solving particular player experience optimization problems (like, for instance, (Yannakakis and Hallam, 2007)), nor supporting designers in solving specific design problems (as in (Yannakakis, Liapis, and Alexopoulos, 2014)); rather, we seek to research how we can contextualize and organize PCG and experience evaluation methods in the game design process.

3.2 Research Questions

We can now sum up our research agenda in the form of two sets of research questions. The first set defines the scope of the problem we aim to solve with the introduction of a novel information system artefact. Each of the set's questions highlight the research path to a set of tool qualities that can solve the research problem. The answers to these questions will be materialized in a tool design proposal – hereupon named the Authorial Game Evolution (AGE). The materialization of the AGE design proposal, the design knowledge that results from it, and the findings pertaining to how it solves the problem, are answers to these questions.

1. *How can game designers use experience evaluation methods and procedural content generation algorithms to prototype video games that mediate intended forms of player experience?*
   (a) *How can we delegate parts of the game design process to automatic generation algorithms, in ways that enhance designers creative authorial control over the resulting artefact?*
   (b) *And what are the requirements for procedural content generation strategies so they can solve game design problems in this way?*
   (c) *How can we combine experience evaluation methods with procedural content generation tools, in a process that can promote designer influence over player experience?*
   (d) *How can game designers interface with procedural content generation algorithms so that they can solve game design problems?*

The second set of questions guided us in studying the impact that the AGE proposal can have on game design processes. Meaning, how well does this tool solve the problem, and how does it affect, both positively and negatively, the process it aims to improve?

2. *How does the introduction of AGE transform game design processes?*
   (a) *With what processes and goals can game designers appropriate AGE?*
   (b) *What trade-offs result from substituting manual game authoring processes with semi-automatic processes supported by AGE?*
   (c) *Can the AGE approach foster creative processes in game design?*

Together, these two sets define the scope and depth of our research. The answers we have found throughout this research process are collected and discussed in Chapter 8.

3.3 Methodology

The main objective of this research is to produce a positive impact in real-world game design activities. Thus, we root ourselves in the philosophical tradition of a pragmatic world-view, concerned with applications and solutions to problems, and not with (post-) positivist approach of finding antecedent causes to effects or outcomes (Creswell, 2013). As per Figueiredo and Cunha (2007), this corresponds to Mode 2 knowledge production, or Designerly knowledge, i.e. knowledge *"needed in a solutions-oriented world: the knowledge to design solutions and purposefully change the world"*, as opposed to explanatory knowledge,

needed for *"analyzing, describing, understanding, explaining, and predicting"*.

The challenge of this research, on a macro level, is in finding a solution – a tool – for a specific game design problem, and then studying its usefulness. From this process, we may find knowledge that can eventually serve as the basis for more robust theory (with predictive power over the end-result), but the focus is not on the production of such a theory. The choice for this methodology is further justified by the fact that at the onset of this research, as far as we can tell, there were no established theories that could support the tool's design so it would achieve its intended goals. Furthermore, if such a theory did exist, there likely would be no need for the construction of such a tool in the first place. As such, the production of a method that can solve the problem we set out to solve cannot be derived from theory or implemented according to any validated model, therefore necessarily relying on a jump into the unknown, a creative step in terms of design. This rationale led us to a Design Science Research (DSR), a methodology that is defined by including a step – an Abduction – as integral part of its research process (Vaishnavi and Kuechler, 2004).

DSR is a complementary approach to positivist and interpretive perspectives for research into Information Systems, which involves *"the design of novel or innovative artifacts and the analysis of the use and/or performance of such artifacts to improve and understand the behavior of aspects of Information Systems"*(Vaishnavi and Kuechler, 2004). This approach intends to understand phenomena that are created, or designed, during the very research process. It also seeks to determine areas of applicability of theories, as the operationalizing of constructs, models and methods into artefacts allows their validation in a given situation or context. The output of the research is then not only the artefact itself, but also knowledge that results from its construction, performance and usage, thus mirroring our goal closely. The main contribution of this research lies in the novel tool design aimed at solving a specific problem. To evaluate how well the tool's design fits the problem it is meant to solve, DSR proposes artefact instantiation as a way of validating the design, for *"artifact instantiation demonstrates feasibility both of the design process and of the designed product"*(Hevner et al., 2004) Then, based on the instantiation, we could *"determine how well an artifact works, not to theorize about or prove anything about why the artifact works"*(Hevner et al., 2004).

According to Vaishnavi and Kuechler (2004)'s DSR process (and due to DSR's focus on problem solving), research begins with Awareness of a Problem phase

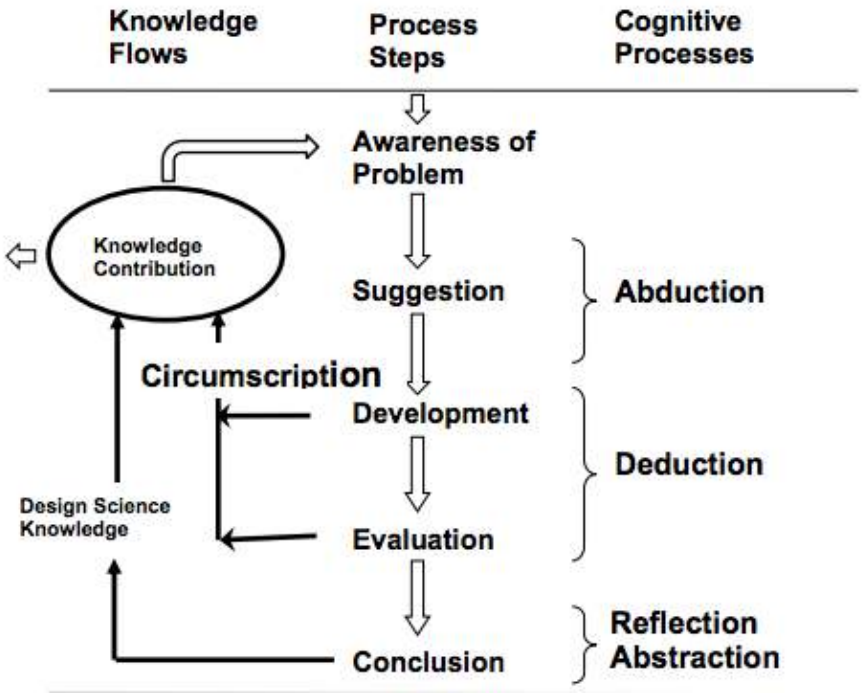

FIGURE 3.1: Design Science Research Phases, as per Vaishnavi and Kuechler (2004).

that outputs a research proposal. During the Suggestion stage, a tentative solution to the problem is proposed; this *"is an essentially creative step wherein new functionality is envisioned based on a novel configuration of either existing or new and existing elements"*(Vaishnavi and Kuechler, 2004). Suggestions are supported by existing knowledge, yet they might be inadequate, or suffer from knowledge gaps, which is why they are the result of a creative process that involves abductive reasoning. The suggestion is then materialized in a complete or partial artefact in the Development stages, and it is then evaluated according to a planned specification during the Evaluation stage.

The process iteratively repeats itself through Circumscription, as researchers revise their problem constraints, assumptions, models, and respective applicability. In this stage, explanatory hypotheses are reformulated, suggesting changes to the resulting design. Research is concluded when results are satisfactory, or deemed good enough for solving the problem, as stipulated. Through reflection and abstraction, knowledge contributions are made, preferably design theory.

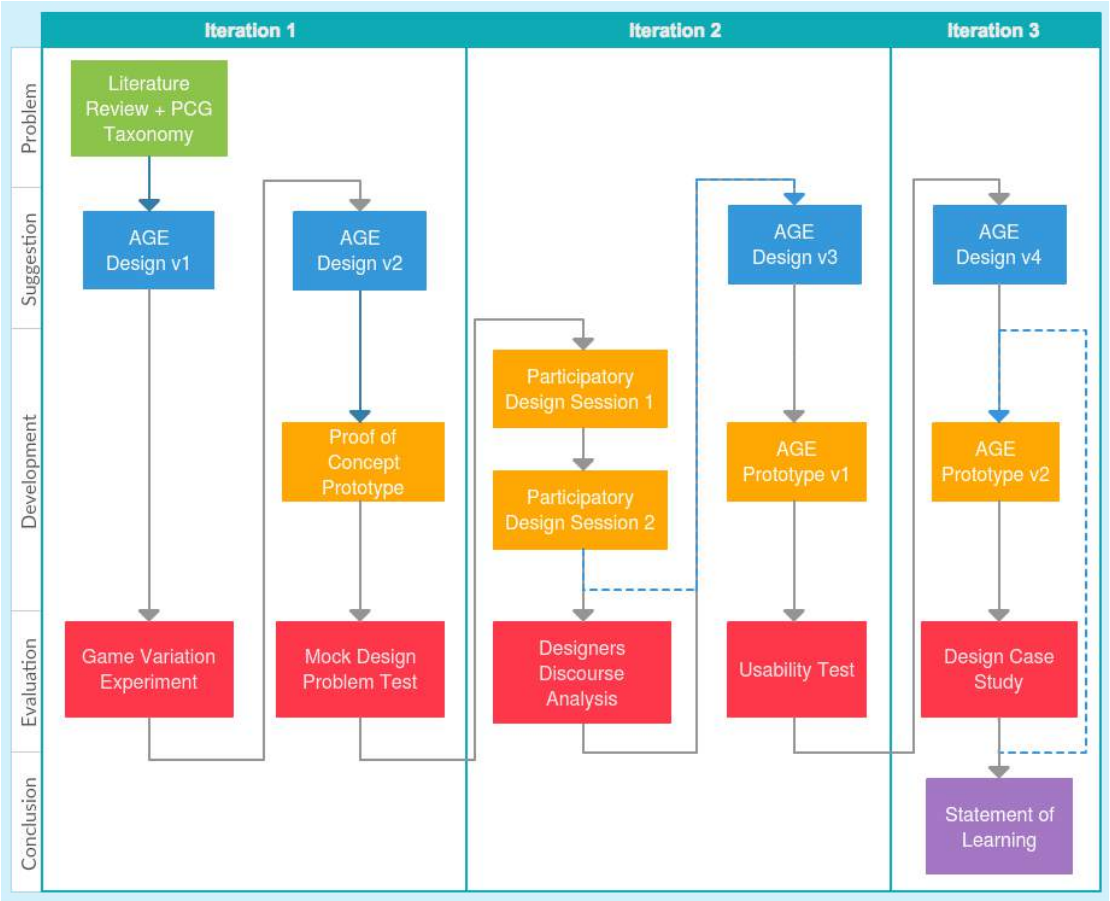

FIGURE 3.2: The research project divided into Design Science Research Phases of Awareness of Problem, Suggestion, Development, Evaluation and Conclusion. Each iteration corresponds to a sequence of phases that is concluded with a a response to a subset of research questions.

3.4 Research Plan Outline

We followed the DSR procedural blue print in the development of this research. Figure 3.2 shows our research process, describing each step as a stage in the DSR cycle according to Vaishnavi and Kuechler (2004). DSR provides an overall research process guidance, but it does not detail which research methods should be used to validate artefacts. Because DSR's contribution arises from artefact's utility, said utility must be rigorously demonstrated during the evaluation stage (Hevner et al., 2004), but the choice of research methods is, for good reasons, highly dependent on the nature of the artefact and its proposed utility. For each of the evaluation stages in our research project we used mixed-methods (Qualitative and Quantitative) approaches to data collection and analysis. This choice is supported by tradition in both pragmatist strands of research (Creswell,

2013) and research into creativity support tools (Shneiderman et al., 2006), an area with which this design proposal dialogues with.

Furthermore, and in line with how DSR typically decomposes research problems into subsets (Hevner et al., 2004), we also employed the term iteration to describe one or more (incomplete) Design Science Research cycles that solve one research sub-problem, each iterating on the previous, and each with its own contributions. This resulted in what we are classifying as 3 distinct Research Iterations which we describe in the next sub-sections. The first iteration concerned the creation of a functional design specification for the AGE tool, the second, the design of an interface for this tool, and third, the study of the tool in use.

3.4.1 Iteration 1 – AGE Functional Design Specification

Phase 1 focused on defining the scope of the research problem and determining the form the solution should have, leading to the validated functional specification of the AGE tool as well as a first proposal for its use process.

Research started with an analysis of the state of the art that resulted in the problem definition; its material counterpart lies in the first three chapters (chapters 1, 2 and 3) of this thesis. Additionally, in order to better understand the PCG domain, its possibilities and state of the art, we proposed a taxonomy of PCG solutions, focused on clarifying actor roles (chapter 4). This was a relevant step in understanding in outlining possible contributions in this area. Specifically, providing this novel way of categorizing PCG techniques and solutions, allowed us to map undeveloped research approaches. Based on this analysis, and through abduction, came the idea for AGE's design in its first incarnation, expressed in chapter 5.

However, when AGE's initial design was outlined, there were doubts on whether certain assumptions underlying it would hold, particularly what sort of procedural algorithm would best fit this tool's approach. At that moment, there was an opportunity to carry out an experiment with an existing game prototype. Before committing to the development of the AGE design, a preliminary test was carried out intent on gathering data that could guide us moving forward (that is documented in section 5.2). Note this trial was not, strictly speaking, an evaluation of AGE as it had not been developed yet – instead being carried out on an adaptation of an existing game prototype that we had at our disposal. We used the game artefact to evaluate some elements in AGE's design, and the results of this evaluation were then taken into account in a design update.

The preliminary experiment was the only step in our research project that did not neatly fit the DSR research process and its underlying logic; however, this heterodox step was needed for determining key aspects of AGE's design and so should be reported. Succinctly, results showed that generated game design variations indeed impacted player experience, but did so with a great degree of unpredictability, both in relationship to a-priori expectations and in terms of variability between players. This lead us to chose a search algorithm that had no assumptions in terms of the search landscape and dealt well with unstable fitness landscapes.

AGE's design was iterated based with this new data: thereon for experiment purposes we used a genetic search algorithm. AGE's design version 2, included a more mature specification of AGE's functional design and expected use process, as well as guidelines on how to frame search algorithms' procedural generation.

Once the new design version was closed, a proof of concept prototype was implemented based on this specification. It includes all the systemic bases of the (then existing) AGE design save for a user interface. The first prototype was functional only – it allowed definition of Design Problems, and a simple genetic algorithm could then try to solve them. To evaluate said prototype, a simulated design problem was established and put to a test in a Mario-like game (section 5.3). The test demonstrated it was possible to find a solution for a design problem using AGE's approach, and did not force any revaluation of AGE's design. This only served as a feasibility check in terms of AGE's functional working; meaning, it was at least possible to solve a simple optimization problem using the AGE approach.

3.4.2 Iteration 2 – AGE Interface Design

The next phase concerned the development of a user interface for AGE; this is outlined in chapter 6. Seeking to make it usable by designers, and improve its potential appropriation by the community, we held two Participatory Design Sessions with several stakeholders in AGE's development (section 6.1). The purpose was to bring in invested designers into the process, and let them craft the interface that made sense to them in a participatory manner. The first output from these sessions was a series of paper-prototypes of the AGE interface. Additionally, data from designers' discourse during these sessions was analysed so as to try and develop insight into how they came to understand and appropriate AGE. From that analysis, several edits were made to that basic prototype, in

aspects which the participants struggled with.

This effort converged in a new version of AGE's design, complete with user interface, a new use-case, and some minor changes to its working. The ensuing design was prototyped into a web-application, and had its usability evaluated. Besides testing the functioning of the application, several solutions to terminological problems (that popped up during participatory design sessions) were evaluated. Once this phase was completed, a full AGE prototype was ready for evaluation.

3.4.3 Iteration 3 – Case-Study Evaluation

Once the new design was completed, a case-study with the first fully developed AGE prototype ensued (see chapter 7). The goal was to do an in-depth analysis to how AGE was appropriated by a designer in a as-close-to-real-life-as-possible design scenario, to achieve a good degree of ecological validity. As the case-study was under-way, minor changes to AGE's interface were made to accommodate the designer's needs, which lead to a succession of design iterations incorporated in the end-design. The resulting contributions of this case-study is the confirmation of the tool's usability by designers, and a wealth of key findings on how it can be appropriated by game designers.

However, because it is only a case study with just one designer, it only demonstrated the potential usefulness of this tool. Therefore, it does not conclude research into AGE, merely stands as sufficient validation of the tool's feasibility and its potential impact in the game design process. Further DSR cycles will be needed in order to provide extensive evaluation of its contribution for game designers and producers, besides further revisions of its design.

Despite this, the thesis concluded at this stage, as both AGE's design and its novel design process were evaluated as being feasible, and considered detailed enough to be implemented, evaluated and further refined by other parties. Furthermore, data amassed during this process allows us to point to new research avenues hitherto unexplored, which require a lengthier research cycle than what was available to us (see chapter 8 for more detail on these matters).

Chapter 4

A Taxonomy for the Clarification of PCG Actors' Roles

PCG is a not a new method in game design practices. Though PCG solutions have been around for decades (Hendrikx et al., 2013), their impact on game design roles and procedures is, we argue, still undeveloped. Recent years have seen a surge of research in this area, but we think the bulk of the effort has been on improving the computational algorithms used to generate content, and on how to better optimize and tailor said content to satisfy players. Not enough study has been directed towards studying how designers can interface with PCG systems.

Furthermore, despite the proposal of several models and taxonomies, there lacks a unified perspective for identifying and cataloguing a PCG system. Particularly, in a way that allows clear understanding of how designers can interact with these systems in a game creation context. Because we are intent on improving the design process by appropriating these computational methods, it seemed natural to study how existing systems allowed human authors to affect the end-result.

Our view is that PCG can be understood in three distinct ways. One, as a computational creativity agent that replaces the human author in the production of a video game or part of its content, a 'PCG designer'. Two, as a tool to used by human creators during video game production, with procedurally generated content being incorporated into a video game, a 'PCG design tool'. And three, as part of a finished video game artefact that features in-game content generation, a 'PCG game', designed by humans with or without aid of 'PCG design tools'. And yet, existing PCG surveys and taxonomies (Togelius et al., 2011; Yannakakis and Togelius, 2011; Hendrikx et al., 2013; Smith, 2014)) do not elaborate on this

distinction, classifying methods without taking into consideration how designers can interact with PCG methods during video game design and production.

Therefore, this work seeks to propose a classification system that clarifies the multiple role configurations that human designers, PCG algorithms and players can have in the context of PCG-aided video game design and production. Towards this end, we will synthesize and redesign existing taxonomies in a way that highlights this divergence. Our aims with this taxonomy were: first and foremost, provide better understanding of how human actors can interface with PCG systems; second, to synthesize existing taxonomies, providing a unified model that had the best elements of pre-existing classification systems; and third, to facilitate communication and classification of existing approaches to game designers, by trying to use a more intuitive terminology.

4.1 Limitations of existing Taxonomies

We believe each positioning for a PCG research reflects a different sensibility, and frames methods in their respective taxonomies accordingly. While the value of these taxonomies is indisputable, we have some reservations to their internal logic, nomenclature and consequent coverage (for detailed descriptions, see section 2.2).

Hendrikx et al. (2013) only factored method and content type, and ended up with a very conceptually narrow Taxonomy, useful in listing existing solutions in the industry, but not particularly powerful in terms of providing a space for classification of innovative PCG methods, or for finding identifying traits on the nature of each method outside of its generation algorithm and output. Take, for instance, the family of Experience-Driven PCG methods (Yannakakis and Togelius, 2011): despite the firm heterogeneity of possible solutions, they would be lumped together and categorized under the AI category, with only their algorithmic basis and content type serving to distinguish between themselves through sub-categorization, when, for example, Yannakakis and Togelius (2011) taxonomy identifies several distinctive characters that inform us on their procedural aspects and that go well beyond simple mentioning of their algorithm. Furthermore, Hendrikx et al. (2013) nomenclature is at times problematic: the aforementioned AI category illustrates this fact, as all other categories make reference to AI algorithms, only of different families. To conclude, we think that methods and output are valid characteristics to form a PCG taxonomy, but are not enough to capture the wide diversity of qualitatively different solutions.

Yannakakis and Togelius (2011) EDPCG taxonomy fares much better in terms of identifying key characters for categorization of PCG approaches. Its division of EDPCG into four distinct components – Player Experience Modelling, Content Quality, Content Representation and Content Generator – with several different traits in each, allows for a finer, more meaningful system for classification and identifying of different solutions. Furthermore, it has an advantage in that it clarifies how players and player-derived data are factored into existing PCG solutions, something amiss in Hendrikx et al. (2013). Some problems remain however. First and foremost, it abandons, or at least, minimizes the relevance of some branches from other similar works by the same authors, namely the online versus offline distinction and the different variants of PCG (Togelius et al., 2011).

There is also a distinct lack of inclusiveness: admittedly, this is a quality inscribed by design. By only addressing EDPCG solutions, a vast scope of different-minded solutions (say, any classical PCG solution with no player interaction) get left either uncategorised, categorized solely in the negative (for lacking any characteristics that conform to the four defining components of this model) or simply poorly defined (when their distinguishing features are outside the four characters). We find at least two striking examples: both Sentient Sketchbook Yannakakis, Liapis, and Alexopoulos (2014) and Tanagra Smith, Whitehead, and Mateas (2010b) involve subject interaction which defies both the Player denomination in 'Player Modelling' (as subjects behave like creators and not players, constructing the end-solution in tandem with PCG systems), as well as its taxa, existing none where a role of active design of the solution can be properly framed.

This brings a crucial question of how would video game producers appropriate EDPCG systems. No hint is given in these papers as to what sort of control of content generation would be afforded to designers and programmers, whether they would have to code their own variant of these systems, adapt them by providing base-content to fill out gaps, or if there would be significant parametrization of these methods. The way EDPCG is proposed, the human designer is not considered explicitly, and it is our position that there is no reason why this should be so. Again, it must be stressed that these are not a problems with the EDPCG taxonomy in of itself, merely a stating that its scope that was never intended to be abusively generalized to the whole spectre of PCG solutions, therefore making it inadequate for the broader categorization that is needed to establish.

Smith's (2014) own taxonomy broadens the inclusiveness of previous efforts, and fixing some issues, while providing greater detail on the distinguishing features of each solution (besides other contributions). However, it still has its share of problematic choices. We start with a repeat of the subject role issue; Smith considers the human designer to an extent previously absent from taxonomies, especially when she addresses the Building Blocks mechanic. Nonetheless, when she considers interaction with the PCG system she only addresses Player Interaction, from whence we surmise that in her model Smith (2014) conceives of the designer only in the creation of the PCG system in of itself, with no interaction of an outside designer with the PCG-system, making the PCG-based game (per her own nomenclature) the end-artefact in of itself (and not a tool for design, as is common in several practices (Hendrikx et al., 2013)).

Confirming our interpretation is the way Smith subtly changes Togelius et al. (2011) taxa which refers to the phase at which generation occurs. While she utilizes the same states under the category "Game Stage" – off-line and on-line – she shifts the off-line term from *"during game development"* (Togelius et al., 2011), to what *"happens before a unit of play experience begins"* (Smith, 2014) which, considering her examples, means after production and release, but before game-play proper occurs, as when players are navigating pre-game menus. Her lack of distinction when it comes to the word "human" with no clear distinction between player and designer also points in that same direction.

To Smith (2014), PCG in games is referred only to games with incorporated PCG systems (PCG in the game), and not games that were produced with PCG algorithms (PCG as design tool). In related work, this distinction is made more clear, as in Smith et al. (2012), PCG-based games are defined as games *"in which the underlying PCG system is so inextricably tied to the mechanics of the game, and has so greatly influenced the aesthetics of the game, that the dynamics of player strategies and emergent behavior revolve around it"*. This definition leaves out not only video games created with PCG design tools, but also substantial part of all PCG-games, as only in the cases where players' game-play actions actively interact with the PCG system they are considered PCG-based games.

There are arguments in favour that the two should not be mixed, but the other taxonomies did engulf these two approaches, and in good reason, as the same PCG algorithms can be appropriated in these two contexts. Furthermore, because of this positioning of hers, designers' role when interacting with PCG are not taken into account.

It is our opinion that these efforts in categorization can be assembled together, expanded upon and improved, with their minor flaws corrected, and their positioning opened, so as to provide vaster, more inclusive PCG for games taxonomy, which is what this chapter sets out to accomplish.

4.2 Taxonomy Proposal

Expanding upon Smith (2014), we aim to establish a vocabulary that facilitates and enriches communication between the different practitioners and researchers of the field. We also wanted, for reasons of our own research, to detail the multiple role configurations of roles of human and computer actors when using procedural content generation, with particular emphasis on the human author, a factor we think that has remained mostly implicit in previous taxonomies. It is our stance that this clarification is a necessary first step in tracing how human creators and procedural algorithms can end up affecting players' experience, and their perception of creativity and quality of the end-artefact.

In order to position the context of this proposal, we must put forward our own understanding of what constitutes Procedural Content Generation for video games, seeing as this limits the scope of our problem. To Yannakakis and Togelius (2011), PCG is the *"creation of content automatically, through algorithmic means"*. We wish to expand on this definition so as to highlight how human creators can interact with these systems, and so we must broaden the automatic clause. Thus, we define Procedural Content Generation as the process by which an algorithmic method can, autonomously or semi-autonomously, be capable of generating video game content.

We define content as any component part of a video game, or a complete or incomplete video game artefact or prototype. Because this definition amplifies the notions of content beyond its original meaning, we will use the term solution to refer to the output of a PCG method, be it content or otherwise. With this definition, we can include all the variants and families of PCG: "PCG designer", "PCG (as a) design tool" and a "PCG games".

This taxonomical proposal results from a process of synthesis and expansion of preceding taxonomies, maintaining, whenever possible, common grounds to them and incorporating their unique aspects, while adding any we found lacking. Complementing the study of these taxonomies, we list some of the state of the art PCG methods in research, and some commercial games that were

created with PCG methods or that included PCG in them, forwarding these as examples of each classification branch whenever needed.

Terms and branches from previous taxonomies were maintained and fused together, though effort was made to adapt, simplify and improve the end result. We sought to strike a balance between inclusiveness, communicability, understandability, and categorization accuracy.

4.2.1 Generation vs. Evaluation

We separate the content-generation act into two moments or tasks: solution generation (proper) and solution evaluation. Because this division is a crucial quality of our taxonomy, and it can be seen as controversial, what follows is a succinct justification of this choice supported by creativity literature and a survey of existing PCG methods. Csíkszentmihályi (1997) studied human creativity, and found that the creative act was composed of a recursive process, laden with repeated loops around a set of five steps.

The steps were: preparation (study and immersion of a creative area's problematic issues), incubation (playing with knowledge of the area, churning ideas around, establishing unusual combinations and connections), insight (the eureka moment, when key pieces of the puzzle fit together), evaluation (when the person must decide whether the insight is valuable and worthy of pursuit) and elaboration (development work on the original insight).

We think three first steps can be analogically understood a generation procedure, more specifically a global search kind of process: you take knowledge from an area (preparation) and try out different, improbable combinations (incubation) until a pattern or a fitting solution emerges (insight). Then, a form of evaluation of the solution's quality has to be made in order to decide on its merits; this determines whether or not work is forwarded to developing said solution (elaboration) (the final step does not have an obvious pairing in PCG methods, though it can be understood as a form of local search, a refinement of a base solution, or even a new generation phase based on previous solutions).

The relevance of evaluation in creative computer systems is, for example, advocated in Machado et al. (2004), where it is argued that a *"genuine evolutionary artificial artist (...) requires (...) a system that is able to "perceive" an artwork, and perform an evaluation of the piece"*. In Computational Creativity systems, that seek

to emulate or parabolize human creative processes, this distinction is very common. Besides *"generating new material"*, these systems include the responsibility to develop or employ the means to *"assess the value of the artefacts"* they produce (Colton and Wiggins, 2012).

Similarly, co-creativity methods, which pair human and computational, include *"the evaluation of the final (or possibly intermediate) outcomes and the evaluation of the co-creative process for the generation of outcomes, solutions, or items"*(Yannakakis, Liapis, and Alexopoulos, 2014). Procedural content generation methods are inscribed in these traditions, and thus tend to reveal this pattern of generation and evaluation. This is evident in generate-and-test and search-based content generation (Togelius et al., 2011), all experience driven content generation (Yannakakis and Togelius, 2011) and constraint-based methods like Smith (2014)'s own Tanagra.

All those methods, besides generating content, also have procedures to either assess its quality or at least validate it. Despite this, not all methods have some sort of algorithmic evaluation. Other alternatives, like constructive and grammar generators, for instance, have procedurally generated content without any form of automatized algorithmic content evaluation.

However, we would argue that there are other forms of implicit or explicit validation of solutions' quality: either pre-emptively inscribed in the generator's algorithm as it was being created - so that it would tend to give results that the human designer found valuable (more on this in the next section) - or have their output evaluated by human designers, that then proceed to either include or discard generated content in their design.

The added value in separating these two phases lies in being able to identify, with greater accuracy how each actor impacts the end-result, as the qualitative aspect of it depends on whether or not they interfere with each phase, and in what role. This also allows us to track, with fine detail, the responsibilities of the different actors in terms of the quality of a play experience mediated thanks to PCG systems. By **Generation** we understand all processes with the goal of creation, recreation or iteration of a game content solution, irrespective of its validity or quality. **Evaluation** refers to any procedures or acts, be they formal or informal, that end up determining the value of any generated solution, in a way that guides the creative generation process in subsequent iterations.

4.2.2 Overview

Figure 4.1 shows a visual representation of the various categories for this Taxonomy. It is a 3 by 2 table, in which rows correspond to the different actors, and columns to the two PCG roles they can enact, evaluation and generation. In each cell lie all the alternatives for a specific combination of Role and Actor. Possibilities, unless otherwise stated are *not* mutually-exclusive, with several possible combinations being possible, if not frequent in existing methods. Note however that one specific cell, **Computer Generation**, is an exception, and has a further subdivision into three sub-taxonomies, each pertaining to a particular aspect of that role-actor combination.

Classifying PCG methods according to this model requires only that a specific question be answered for each combination: how does the Actor (Designer, Computer and Player) impact the Role (of Evaluation or Generation) in the context of this PCG method? If one fills out the type of impact each actor has or lacks in these six combinations, than the PCG methods' interaction with the human actors in a game design context is made clear. The following subsections have descriptions of each branch and the definition of each categorization possibility.

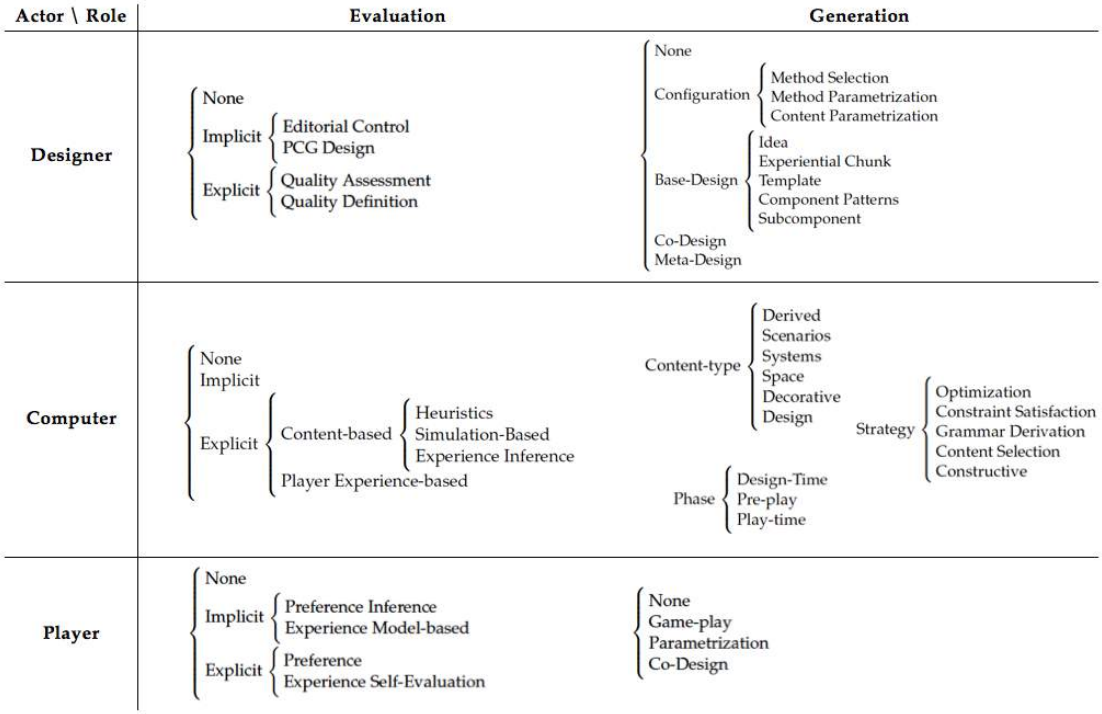

FIGURE 4.1: Taxonomy used to clarify actors roles in PCG methods.

4.2.3 Designer x Computer x Player

The main addition of this taxonomical model is the direct and explicit enunciation of how Human and Computers interact in PCG powered design contexts. This serves to further distinguish between different PCG alternatives, as methods in which PCG algorithms act as the primary creative force in the process will have a greater prominence of the computational role face the designers', whereas co-creativity alternatives and PCG as tool alternatives will be more balanced.

By **Designer** we mean all human creators that directly impact the design of the game during production and before it is released to the public; this includes everyone that interacts with the PCG system, from producers, game designers, level designers to game programmers. This branch then seeks to enumerate all content that is created by human beings that interacts with the computational algorithm, which can then alter, expand, blend, and reconstruct it; it also includes the impact humans have in the creation and/or use of the PCG system itself.

It is our opinion that an excessive focus on Players and Player Experience Modelling has left Designers' role in the use and creation of PCG solutions sorely undeveloped. By attributing a role to their participation in PCG enabled design processes, we clarify and make explicit their impact in PCG solutions, and establish a clear path for communication between researchers and designers.

By **Player**, we mean any human who plays the game after its release. Note that in some cases ("Minecraft" (Persson, 2011), "Dwarf Fortress" (Adams, 2006)", "Little Big Planet"(Healey and Smith, 2008)), players can interact with the game in a way that changes its experiential content, be it thanks to level editors or procedural content generators that come with the game, making their role seem akin to that of designers; despite this, for the purposes of this taxonomy, they are players and only players. Player experience data has an especially big role in EDPCG solutions (Yannakakis and Togelius, 2011), and therefore needs to be taken into account.

We use the term **Computer** to refer to all algorithms that are executed either during game production or during game-play, that with complete or partial autonomy, can generate game content. This excludes any computational tool which is directed by a human and has no end-result autonomy, such as modelling applications, level editors, etc.

4.2.4 Designer x Evaluation

In this section we list all aspects in which a designer can impact the evaluation processes of a PCG enabled video game. This includes any direct or indirect manipulation that affects autonomous computational evaluation methods. **None** refers to examples where the computational agent does not take into account any Designer feedback into evaluation processes; the aesthetic evaluation of resulting solutions is decided by computational algorithmic processes and/or player input. For the Designer to have no impact in this category, it is also absolutely necessary that he was not the designer of the PCG solution. In this latter a case, we use the **Implicit** categorization: this means any way that the designer might indirectly affect how solutions are evaluated by the designer-computer-player complex. Finally, **Explicit** refers to all the possible ways in which the designer overtly impacts how solutions are evaluated.

The basic and most common form of implicit Designer Evaluation is **Editorial Control**. When designing aided by any tool, creators can discard partial or complete solutions at any time; the same is true for procedurally generated design solutions. Irrespective of the quality attributed by computational algorithm and/or players, designers thus reveal an implicit act of evaluation on their part. One notable example of this form of qualitative control occurred during the development of Elite (Brabben and Bell, 1984), where Brabben and Bell tried out several sets of possible seeds and universe sizes in order to find an interesting compromise between size and quality of generated systems (Spufford, 2003). Though we did not find further examples of this, it is highly likely that when using PCG tools in development phases, creators will often try and repeat procedural processes, so as to obtain the best results.

Another alternative we find is the one in which the designer takes part in the process of designing the PCG system itself, **PCG-Design**. If creators of a game program their own PCG method and code its procedures in a way that it somehow implicitly promotes certain qualities in generated solutions and demotes others, then this is an implicit form of solution evaluation impacted by designers, even though no formal evaluation method may be held. This is potentially the case if designers program Constructive, Grammar algorithms, for instance, which have no computational evaluation method, but tend to seek out specific kind of solutions as if they were (implicitly) evaluating content, and pursuing it.

Explicit Evaluation refers to cases in which designers are the ones attributing

quality to generated solutions. The most direct approach is designer *Quality Assessment*, which are those cases in which procedural generation methods present solutions to designers so they can then evaluate it in some way, by either selecting which solutions to pursue or which quality value is attributed. Interactive evaluation is common in evolutionary art (see (Takagi, 2001) for a survey on several), though less frequent in respects to games (Yannakakis and Togelius, 2011). Examples include a meta-generator of level generators (Kerssemakers et al., 2012), where users evaluate a generator via its generated level sets, and a system for generating game buildings where users define which (high-quality) buildings are evolved (Martin et al., 2010).

Quality Definition refers to cases in which the designer does not attribute quality to solutions himself, but defines the way in which the PCG system will automatically assess each solution's quality. The added value of this branch is allowing designers to automate the process, while not losing control over it. In evolutionary art, we know of at least two examples in which human users can *"express their intentions through the design of fitness functions"* (Machado et al., 2014; Machado and Amaro, 2013).

We are not aware of any PCG approach that gives designers the power to map-out how evaluation takes place indirectly, though considering how many PCG solutions use Experience Models (see (Yannakakis and Togelius, 2011)), there is no reason why these should not offer the possibility to establish multiple levels of configuration and manipulation by designers; and in fact, Togelius et al. (2011) assumes this possibility of evaluation function personalisation in their work. As a simple prototypical example, imagine a PCG solution for a Mario level generator where designers could define a weighted sum fitness function, attributing weights to content's classification as fun, challenging or frustrating by the models in (Pedersen, Togelius, and Yannakakis, 2009); different designers could thus express their preference in terms of design through distinct combinations of weights.

4.2.5 Designer x Generation

This branch refers to all creative impact that a designer can have in the production process of a PCG powered video game, which can be related to the use of PCG techniques.

The designer can have four major types of generation impact. **None** corresponds to cases in which the computational system would create, with absolute

autonomy face the designer, every aspect of the video game content in question, without using any designer-authored content. Also, the algorithm would have to show independence and unpredictability face the games' human designers, i.e. having been programmed by someone other than them. Though we could find no prototypical case today of this type of PCG as designer, there is no reason why the possibility should not come to pass.

Configuration refers to when the games' creators interact in some way with a pre-programmed PCG method that offers some degree of customization to its operation. Three sub-types of impact were identified. **Method Selection** is for when a particular PCG tool offers different methodological alternatives to its generation algorithm. One notable example outside of video games is landscape generator Terragen (Software, 2016), which allows users to chose which algorithm to employ in land and cloud generation (Subdivide and Displace, Perlin Noise, etc). **Method Parametrization** refers to when procedural aspects can be configured in the generator, for example, include a system for generating game buildings where users can parametrize genetic operators' strength and likelihood (Martin et al., 2010).

Content Parametrization goes beyond just selecting or parametrizing the algorithm type, being reserved to PCG approaches where the output is functionally dependent on a set of input parameters, and these offer *"clear and predictable result for the output"* (Smith, 2012). **Parametrization** is available in Terragen (Software, 2016), as several features that determine the end-result can be selected a priori (percentage levels of Realism, Smoothing, Glaciation, Canyonism). Another example from evolutionary art comes from (Machado et al., 2014), where *"control is given to the users by allowing them to specify the rendering details of selected pieces"*. We found other examples for video games with **Content Parametrization**, but these were for player use and will be listed accordingly, though the logic is quite the same. Grome, a terrain generator (QuadSoftware, 2016) is an interesting example of PCG as it allows all three types of configuration. Naturally, the more configuration options are given to designers, the greater the impact they can have in generated content.

Base Design is the branch where human-authored content that computational generation methods need to find new solutions is listed. Quality of end-artefact can be positively affected by high-quality of base design elements

(Colton and Wiggins, 2012) which is why proper identification of human authored elements is so relevant. All known PCG systems use some sort of human-authored basis on top of which they will operate; though more automated systems use less. Base elements can be of multiple types, but we adapted a selection of classification terms from Smith (2014)'s 'Building Blocks' mechanics category, while adding **Idea** for cases in which designers forward high-level concepts as basis for generation; examples of PCG systems that work like this are ANGELINA (Cook and Colton, 2014) and the Nelson and Mateas (2007) system. In the former case this is just a word (or multiple words which the system then derives a single word), and in the latter it uses verb and/or noun. These are then conceptually incorporated into the generation system which abides by them as themes.

To Smith (2014), **Experiential Chunks** are large building blocks for content designed by human authors, that can by themselves, be experienced by players; **Templates** are a generalized form of experiential chunk where human designers left 'blank' spaces for the computer to fill in; **Component patterns** are human designed components that are small enough that by themselves do not greatly impact player experience; **Subcomponents** reference generators that operate at the subcomponent level using the same building blocks as a human designer would. For further detail and examples, please refer to Smith (2014).

Co-design is for when PCG methods assign both computational and human actors equal preponderance in the creative act. Examples such as Sentient Sketchbook (Yannakakis, Liapis, and Alexopoulos, 2014) and Tanagra (Smith, Whitehead, and Mateas, 2010b) are paradigmatic examples approaches where Designers are co-creators, designing new content on top of procedurally generated content and then feeding the generator those human-authored solutions, in an iterative creative process.

Finally, **Meta-design** refers to the design production cases where designers create their own custom-made PCG algorithm, so that it then in turn generates game's content. It is a standard practice in commercial ventures that game creators program their own PCG system (e.g. quest generation in "Skyrim" (Wikipedia, 2016c), map generation in "Minecraft" (Wikipedia, 2016a); see (Hendrikx et al., 2013) for further examples). Because it is equally common in some content types to see producers use general purpose methods that were not created by them (like Speedtree, a foliage generator (IDV, 2016) and Grome, a terrain generator (QuadSoftware, 2016), both outsourced to game studios), this

category serves to distinguish these cases. The **Meta-design** class is specific to the former, as it underlines greater designer control in the generation process.

4.2.6 Computer x Evaluation

This division refers to all solution evaluation processes that are carried out automatically and autonomously by the computational agent. Note that many of the possible methods in this section assume human input of some kind, so this branch should be read in conjunction to both Designer and Player Evaluation.

None: PCG methods might not have any computational impact on content evaluation. Hypothetically, this would be a case in which all content evaluation would be carried out either by designers in production-time or by players in play-time, or in cases where the generator affords no means of content evaluation. It is possible that the PCG algorithm may bias the generation process by some **Implicit** process that betrays a form of evaluation. Though we could not find any overt example of this effect, we think it is bound to be a PCG feature in some cases and this classification should be kept open for it. The closest to an example we could find was "Spelunky" (Yu, 2014) where there is no evaluation procedure, but the way the PCG assembles levels seems to be biased towards convoluted, complicated architectures.

The most explored category is, naturally, **Explicit** forms of Computer Evaluation where the computational agent enacts some formal act of content quality judgement. Two types of explicit evaluation carried out by computers, **Content-based** and **Player Experience-based**.

Content-based refers to any evaluation that takes into consideration intrinsic aspects of the generated content. Say the system is generating a platform game level, if the evaluation module is looking at qualities like the numbers and placement of enemies, pits and other obstacles, and then using these elements as indicators of quality of the solution, then evaluation is explicit and content-based. In terms of Yannakakis and Togelius (2011) taxonomy, this is the equivalent to their Direct Content Quality assessment.

Also somewhat akin to Yannakakis and Togelius (2011) work, we propose to sub-divide **Content-based** evaluation into three types: **Heuristics Simulation-based** and **Experience Inference**, which are identical to Yannakakis and Togelius (2011) Theory-Driven, Simulation-based and Data-driven categories, respectively

(changes in names are meant to open-up, simplify and make each category's meaning clearer). **Heuristics** is for when evaluation functions map a functional relationship between intrinsic features and content quality. This is typically based on some theoretical basis as per Yannakakis and Togelius (2011), though there is no reason why Designers cannot establish ad-hoc functions of this type.

Simulation-based refers to cases when evaluation is based on simulated game-play data, extracted from artificial agents walkthroughs. Finally, **Experience Inference** is for when content qualities are mapped as being predictive of affective states or other experiential qualities, and thus used as basis for content evaluation. **Player Experience-based** also follows Yannakakis and Togelius (2011) work, and refers to when the PCG system evaluates solutions not based on the intrinsic qualities of the solution, but on actual player experience data that is then transformed into a measure of solution quality. For further detail on the previous categories, and examples, please refer to (Yannakakis and Togelius, 2011).

4.2.7 Computer x Generation

It is in this branch where the bulk of PCG occurs: in the automatic algorithmic processes that permit computers to generate new content and new artefacts. As expected, Computer generation hosts the bulk of possibilities. We follow Hendrikx et al. (2013) in the sense that we believe that there must be an identification of both the algorithm family used to generate solutions (what we name **Strategy**, as in the generation strategy used) and the nature of the generated content, the **Content-type**. Also, we add **Phase**, to identify when generation occurs, as per (Togelius et al., 2011).

4.2.8 Content-Type

We followed Hendrikx et al. (2013) classification of single-layered content types, but adapted it somewhat for easier reading and more accurate categorization. **Derived**, **Scenarios** and **Space** were maintained as is and their meaning remains untouched. The class Bits name was replaced by the more meaningful term **Decorative** (taken from (Smith, 2014)), but the original definition remains, as elementary units of game content that the player, usually, does not directly engage with; these include cosmetic elements such as textures, sound and vegetation, but also simple behaviour (i.e. the way objects interact with each other and the environment).

Hendrikx et al.'s 2013 Game Design layer is here replaced with one of its subcategories, **System Design**, as the modelling of real-life phenomena represented in the game, such as ecosystems and entity behaviour. This change was based on two reasons. First, the words game design tend to be used at a broader scope level, as in meaning the entirety of a video game's design, making it a poor choice of terms. Second, it conflated two heterogeneous types of content, **System Design** (referring to *"patterns underlying the game and game rules"* (Hendrikx et al., 2013)), and World design, meaning *"the design of a setting, story, and theme"* (Hendrikx et al., 2013). We argue that examples of the latter can be separated and included in the **Scenarios** content, seeing as that has a subcategory for Story content. **Scenarios** describe the way and order in which events unfold, typically story-related content, but also level sequences and puzzles.

Space refers to the environment where game-play takes place in, mostly map and terrain geography. Finally, **Derived Content** is all content that is a side-product of the game world, such as news and broadcasts which communicate game-events and player leader-boards. For examples and further clarification on these categories, refer to (Hendrikx et al., 2013).

Besides these, PCG methods can go beyond just generating content, and can create (complete or partial) **Design** prototypes, as well as fully fledged video games. These will typically assemble or operate on several different layers of content, and can operate with different generation methods; the most notable research example is ANGELINA (Cook and Colton, 2014). We then reserve the term **Design** for similar cases, as it is makes classification more clear.

4.2.9 Strategy

Strategy answers the question of How the computational agent proceeds to generate content. In this, we followed Smith's categorization, mainly for her better choice of representative and easily communicable classes, so for further examples and clarification of these categories see (Smith, 2014).

Smith (2014) proposes **Optimization**, **Constraint Satisfaction**, **Grammar Derivation**, **Content Selection** and **Constructive** approaches. **Optimization** methods *"treat the design process as a search for the combination of elements that best fit some criteria, which can be either specified mathematically by the system creator, or judged and curated by a human"* (Smith, 2014). Interesting examples include (Togelius, Nardi, and Lucas, 2006; Yannakakis and Hallam, 2007), and all methods that use search-based algorithms and similar approaches.

Constraint Satisfaction approaches are based on the declarative specification of both the properties and constraints that define the content that is to be generated. Examples include Facade (Mateas and Stern, 2003), Tanagra (Smith, 2012), and on the commercial front, games like "Left4Dead" (WikiA, 2016). Grammar approaches are when content is generated by an expansion of grammar-like rules. **Content Selection**, a humble approach which relies on the simple act of sampling content (from a larger database) and then presenting it to the player. Though assuredly debatable, this category has been proposed by Smith (2014), and has its roots in commercial games. For instance, it was used with notable effect in "LSD: Dream Emulator" (Sato, 1998; LSDWiki, 2016). **Constructive** generators build content by assembling with a predetermined process previously constructed building blocks.

4.2.10 Phase

Understanding when procedural generation occurs is crucial in our understanding of the role of Designers and players. Togelius et al. (2011) defined this with two categories, 'Online versus Offline', that distinguish whether generation occurs during game development, or while players are playing games. Smith (2014) however, seems to have a different interpretation of these terms, as she utilizes them and she shifts the off-line term from *"during game development"* (Togelius et al., 2011), to what *"happens before a unit of play experience begins"* (Smith, 2014) which, considering her examples, means after production and release, but before game-play proper occurs, as when players are navigating pre-game menus. To provide some systematization of the phases in which PCG occurs, in a way that bridges the gap between the different taxonomies, we propose a change from an Off-line vs On-line dichotomy, to a three-way branch: **Design-time**, **Pre-play** and **Play-time**.

Design-time refers to Togelius et al. (2011) off-line, a phase in the development cycle of the video game where it is still not finished. PCG can be used in this phase as either a tool for development of a non-PCG powered game, or as part of PCG-powered game (or, more likely, a bit of both). In this phase, player data may be collected as part of creating an intrinsic evaluation function, but apart from that specific case, we consider players do not interact with the PCG system.

Pre-play refers to our interpretation of Smith (2014) off-line variation - referring to when procedural generation occurs in a released game, but before

players game-play experience in itself starts; examples range from "Civilization IV" (Meier, 2005) to "Dwarf Fortress" (Adams, 2006), etc. **Play-time** refers to on-line generation: when a released game has content that is being generated as game-play occurs, maybe even as a consequence of how it occurs. Commercial examples include 'Minecraft" (Persson, 2011), which has on-line generation of landscape data for issues of load balancing, "Borderlands"' weapon generation (Armstrong, 2009), "Skyrim's" (Howard, 2011) Radiant Quest system, amongst others. For research examples of both off-line and on-line variations, see (Yannakakis and Togelius, 2011; Togelius et al., 2011).

4.2.11 Player x Evaluation

Player actions with finished video game artefacts that affect how generated content is evaluated fall in this branch. Like before, the following three categories refer to whether nor not there is player evaluation, and if there is, if it is an **Explicit** act on their part (usually, but not necessarily, measured by some form of questionnaire), or an **implicit** form of evaluation derived from their behaviour (usually, but not necessarily, analysed through game-play metrics). **Implicit** types include two possibilities **Preference-inference** and **Experience-based**.

Preference-inference refers to when some player action or behaviour betrays their liking a particular quality in generated content, which generators can afterwards map to a value in their evaluation procedures. In (Togelius et al., 2011) an example is forwarded of a PCG shooter by Hastings and Stanley (2010) where a weapon generation system evaluates content based on how often players fire them. In this case, evaluation is not based on any experience model, but on the logic that using a given item betrays preference, and generation should promote solutions identical to those that players like.

Experience Model-based is for when behaviour is mapped to some experiential concept - an emotion, an idea, a type of behaviour, for example - and this can then be used as basis for new content (see, for example experience models in (Pedersen, Togelius, and Yannakakis, 2009)). Commercial examples include "Left4Dead" Booth (2008), where a director AI controls game events and seeks to modulate player tension, stress, pacing, conflict and difficulty (WikiA, 2016).

Explicit refers to when PCG systems use subject-data - typically in the form of questionnaires - to determine how to appraise generated content and generate new content. When players get to evaluate the content they are playing in terms

of their quality (attributing a score to content, or valuing content by comparison of different alternatives), we say that players contribute in to evaluation in terms of **Preference**. We have found no example of this, but it seems an obvious path to explore.

Experience Self-evaluation is for cases in which players evaluate their own experience, but do not judge it in terms of what they want or prefer, so the generator can adapt its output to players affective state. These distinctions between Preference and Experience is may give rise to contention. We propose them because they offer very different design mind-sets and can be used in different design contexts. Whatever data-source used, one serves to maximize player satisfaction, the other to adapt and modulate players experience.

4.2.12 Player x Generation

This branch refers to every player interaction with a video game that can impact content generation.

None refers to PCG methods which do not take into account any Player actions in order to generate new solutions. **Game-play** refers to when **Play-time** PCG takes into account in-game interactions so as to select which type of content to subsequently generate. These are not actions used value game content, nor represent player attempts to determine generated content, but cases where PCG-games react to player actions so as to reply with dynamic consequences (usually **Systems** content generators that view the maintenance of a believable and complex game-world). Examples include "Blade Runner" (Castle, 1997) and "Facade" (Mateas and Stern, 2003), where AI systems determine narrative elements player actions, "Skyrim's" (Howard, 2011) procedural quest generation system that takes into account players' previous actions.

Parametrization (per (Smith, 2014)) is analogous to Designer's **Content Parametrization**. Examples include map generation Civilization IV (Meier, 2005) where players can determine map's topological features, or in Dwarf Fortress (Adams, 2006) where they can determine key game-play map elements (history, size, number of civilizations, sites, beasts, savagery and mineral occurrence). Another example is "Murder" (Harrison and Kingsley, 1990), a procedural murder investigation game, where an initial screen (in the form of a newspaper article) allows players to parametrize several aspects of the experience (names of characters, murder date and location), which then impact the resulting murder mystery's features.

Co-design is for cases in which players can create or edit content in conjunction with the computational agent. Though there are no commercial examples we could find of this type, Tanagra (Smith, Whitehead, and Mateas, 2010b) and Sentient Sketchbook (Yannakakis, Liapis, and Alexopoulos, 2014), if incorporated in a finished game and opened up to players, would be of this kind. Given the prominence of in-game level editors, their integration with a PCG system acting as a co-designer seems like a logical next step.

4.3 Discussion

This taxonomy synthesizes existing taxonomies while solving some of their contradictions and producing a more intuitive set of classes for communication to members outside the research community. Despite the lack of some classes and a reorganization face previous efforts, we think it provides a more effective blueprint for classification of PCG methods. The only downside to this exercise is that the comprehensive nature of the Taxonomy makes it somewhat large and convoluted.

Thus far in this research area, there has been a great deal of focus both on novel methods for content generation and how these can better fulfil player needs, without forming a clear picture of how designers can fit in with these systems. This model, by clarifying all the potential roles human authors can have, both in evaluation and generation procedures, allows us to map out more clearly how designers can influence PCG's output.

We believe that for PCG methods to yield results, they must be designed, first and foremost, with human factors in mind, as there is great potential for humans and computers to complement each other, minimizing each others' flaws and potentiating their creative contribute. Towards that end, more study needs to be directed at understanding how humans can interact with these systems, and this taxonomy aids in that regard, providing a way of classifying PCG methods in respects to what types of control they permit to human designers.

To conclude, this role clarification shows that there are many undeveloped areas of research, as we could find very few PCG methods for video games that fit some of the new classes in this taxonomy. Namely, systems with Explicit designer-centred Evaluation (both in terms of Quality Assessment and Quality Definition), forms of Designer Generation Configuration (particularly Content Configuration), and last but not least, Co-Design approaches that can capitalize

on human designers' creativity.

Based on these ideas, the solution to our Research Problem would revolve around a PCG approach that placed designers in the forefront of the systems' design considerations. We wanted to propose a system that allowed designers strong impact in the end result, affording greater degrees of control through interaction in both Evaluation and Generation phases. The following chapter will detail this proposal.

Chapter 5

AGE Design Proposal

The goal of this research was to find a way in which to improve game design processes, specifically in the area that concerns player-experience design and evaluation. Our approach was to question if there could be a novel way of appropriating existing player-experience evaluation methods and PCG algorithms towards that end.

Specifically, we propose to take procedural content generation procedures, while trying to remove some of the Artificial Intelligence mindset – focused as it is on the replacement of the human actor – and route these towards a process where authors have greater creative control over the output, whilst simultaneously trying to make the process more effective.

This chapter details the initial design proposal for what a tool that achieves should be like. This includes a functional detailing of its procedures (section 5.1), a preliminary experiment (section 5.2) meant to help decide what sort of algorithm could be used for generation, and an experiment designed to study the feasibility of this design (section 5.3).

5.1 AGE - The Authorial Game Evolution Tool

To conceptualize procedural generation algorithms as tools that serve authors' design intent, PCG should be transformed into a tool tasked with solving predictable, mechanical sub-processes involved in the production of games, as these can be automated without loss of designer control, and also because this is what computer algorithms excel at (in opposition to humans). This would also serve to leave the high-level vision of the end artefact (the design in of itself) intact in the hands of the human author, promoting creativity in an

author-centric, designer perspective. Thus, we designed a problem-independent tool that uses automatic player experience indicators measurement and testing tools, and PCG-like optimization of artefact parameters, all according to author specification.

In short, AGE is a tool that uses PCG to solve design problems as defined by authors. In this metaphor, designers are problem-setters – they are in charge, defining the design problem to be solved, its constraints, and how solutions are to be evaluated – while PCG acts as a problem-solver – a worker-bee that explores the solution-space, finding new solutions and assessing their merit, all according to the boss's instructions. To explain how AGE works, we start by outlining the rationale behind its design – namely, its conception first and foremost, as a PCG tool for game design.

5.1.1 Turning PCG into a Tool

The solution to our research problem, in our positioning, is to remove PCG algorithm's autonomy, and place human authors at the helm of the generative process, framing PCG merely as a tool that does what authors bid them. A tool is a

> *"device for making material changes on other objects, as by cutting, shearing, striking, rubbing, grinding, squeezing, measuring, or other process. A hand tool is a small manual instrument traditionally operated by the muscular strength of the user; a machine tool is a power-driven mechanism used to cut, shape, or form materials such as wood and metal. Tools are the main means by which human beings control and manipulate their physical environment."*(Britannica, 2008)

What is crucial to retain about this definition is the emphasis on the key expression *"means by which human beings control and manipulate"*. The driving agent, the primary impactor of the process, is the human creator. One understands the tool, whether manual or machine, as a material extension of the human agent's will. This is not to dismiss tool's impact on the creative act, only to illustrate the boundary that separates tools from authors.

Take painting is an analogy. Painting is the act of intently applying coloured material to surfaces. One can paint on several surfaces: canvas, frescoes, wallpaper or walls proper (as with graffiti art). Any utensil and instrument with

which a painter applies the paint is a tool: spatulas, ink pens, oil brushes, air brushes, knives, spray-cans, etc. These are active tools: they directly interact with the material object that is being created. But one should also consider other types of tools: those that support the creative act, even when they are not interacting with the medium directly. Rulers, compasses, lead pencils used to outline images or drafts, easels, palettes, *et cetera*, do not change the appliance of paint per se, but enable the act in a way that changes the creation process and consequently impacts the end-form of the painting. Take the simple example of a palette: it allows painters an easy way of mixing pigments and producing particular chromatic effects that would be difficult to achieve otherwise (a dégradé for example).

Tools and materials share a number of qualities,

- can make artefacts easy/difficult, possible/impossible to craft (on account of their physical and technical limitations);
- may demand a given skill-level and/or effort to use and implement;
- can only help realize artefacts (partially and completely) with an author's active participation and supervision during the creative act;
- their use leads to (tentatively) predictable and repeatable end-results;
- are passive in terms of what is created with them. A tool has no mind of its own, it is class/genre/format agnostic, even if it might ease the creation of artefacts with certain given qualities, it is always up to the creator to define if and how to apply the tool to achieve a given form.

Tools' and materials' affordances define the space which the creative painter can navigate. He cannot escape their physical limitations, no matter his level of skill or vision, though he can circumvent these by selecting (or in extremis, creating) other tools and materials. *"When the technology changes, the possibilities for meaning making change as well"*(Murray, 2011); change the qualitative nature of available tools and materials, and so the affordances change opening up or closing potential areas for exploration. The history of painting shows this pattern repeating itself. For example: in the 15th century, oil replaced egg-tempera as the basis for paint; because egg-tempera dried quickly, painting had to be fast and precise, with less room for errors and without the possibility of mixing more layers of colour (Gombrich, 2009). In the 19th century, artists' colour palettes roughly doubled (Katz, 1995), widening the chromatic spectrum significantly with the addition of new pigments and bases. And thanks to the introduction of flexible palette knifes and metal-ferruled flat brushes, thick impasto and sculpted

paint became widespread (Katz, 1995), allowing paintings a more textured and three-dimensional quality. Outdoor painting became feasible thanks to premixed oil paint in tin tubes (Katz, 1995), in effect paving the way for the arrival of the 'Impressionist' movement, which stood on the premise of painting the natural world by way of in-situ observation (Gombrich, 2009). The list could go on.

Tools, however, do not have a mind of their own, they can only be used by creators, who utilize them with *intent* towards a specific effect. Tools and materials, though requiring author skill to employ, realize an effect that is largely planned for, and despite there existing differences between the artefact as planned and as realized, there is a small degree of randomness and unpredictability to the process. Despite tools mediating authors' ability to mould materials, and materials considerable effect in shaping the creative act, creative authorship has been set firm in the hands of authors.

"Just as telescopes, microscopes, and cameras are powerful devices that enable discoveries and innovations, they are still only tools; the act of creation is carried out by the users" (Shneiderman, 2007). And just as *"as Galileo and Thomas Jefferson employed and pantograph, contemporary innovators use computer- based software tools"* (Shneiderman et al., 2006). This is because the only actor in the process is the author himself; it is human volition that determines the form of the artefact and human volition alone. Despite tool's severe impact on creativity, tools are never in command of the process. They are, in a sense, unwilling, passive, inert...in summary, uncreative. A brush does not work without a painter holding it in a specific manner; it is he who is in control, who tailors all aspects of the artefact to his vision, from inside the confines established by the interaction between tools and materials.

Let us take this perspective into the computational realm. Imagine we address this using the search-space metaphor frequently used in optimization procedures. The physical constrains of materials and tools define the search space boundaries. There is a cost to the exploration of each area, resulting from imposed biases, soft constraints, or ease of use of particular types of expression. The search algorithm – the way space is navigated, with mutating iterations of the artefact – however, is the sole responsibility of the author. He defines which areas to explore, how to navigate from one to the other, which path the takes from inception to conclusion, including all editorial options taken along the way. Naturally, this search process is heavily influenced by his context: personality, taste, social and cultural backgrounds, available materials and tools,

accumulated technique and experience, etc.

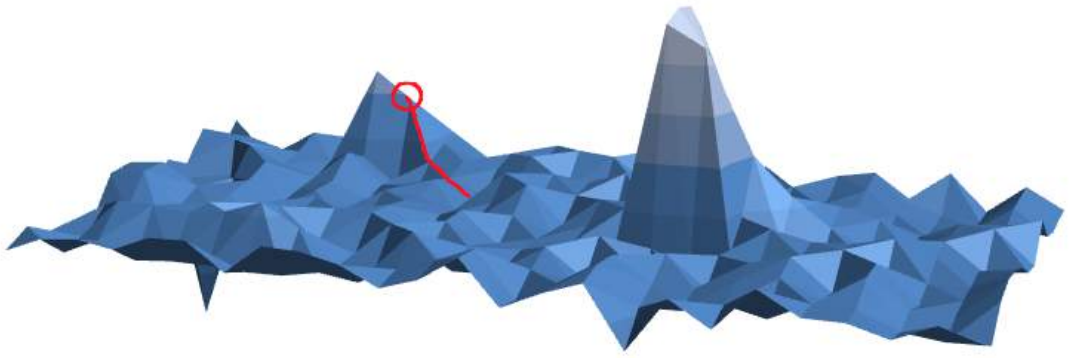

FIGURE 5.1: The search landscape can serve as a metaphor for creative act. The axial boundaries of the graph represent physical limitations imposed by the interaction between tools and materials, and the height represents how costly (in terms of logistics, skill, effort, etc) a given end-result is. The red line symbolizes creators' iterative path through the creative landscape until he finds himself with the end result – the circular red line. This pictorial metaphor aids us in understanding the nature of a procedural tool: if it guides the process, determining the red path and final goal, then it is taking integral part in the creative act. If it impacts on the geographical borders of the space or cost function value (the height of the blue surface), then qualitatively, it is aiding the creative act of the author.

PCG systems can not be neatly constituted as a tool. For it is no longer only the search space boundary and cost-function that are constrained by them, but the actual end result of the search process, given as PCG systems autonomously navigate the design-space when creating solutions, according to their algorithm. The majority of PCG methods we have researched do not allow designers to control the process, or allow for shallow forms of control. Our proposal is to contextualize to put the human designer in charge of controlling the PCG method as if it were a tool.

We are not alone in thinking PCG systems can be complemented and improved by human authors. In co-creativity approaches (Yannakakis, Liapis, and Alexopoulos, 2014; Smith, Whitehead, and Mateas, 2010b), human authors can still interact with these tools so as to collectively design an end-object, with human and computational authoring bouncing back and forth, as if two authors collaborated with each other (Lubart, 2005). This seems a natural path for improving results, but still allows PCG algorithms still an active creative part in the

realization of the end-artefact. The question is: can we go beyond collaboration? And can we do so while maintaining PCG's utility?

We seek to channel the advances these tools afford into a context where they serve human authors' vision, and not replace or collaborate with them. Hendrikx et al. (2013) affirms that *"to avoid losing control over the design process, it is desirable that artists and designers can still influence the final product by adjusting the parameters of the procedure"*. To take methods designed to replace authors and shape them into creative tools for designer use – this means finding ways in which authors can control the effect of these tools, reinforcing all possible aspects of author control over the end-result, avoiding unpredictability and randomness in results, and diminishing to a minimum the active role of procedural generation.

Granted, this seems almost at odds with the autonomous nature of PCG algorithms. In effect, this may seem a contradiction in terms; for if one renders PCG into 'uncreative' tools, then they lose their intrinsic value. To give PCG a Human-centric logic, the design of this tool must then offer a context which balances the intrinsic value added by PCG techniques, while simultaneously afford degrees of control an predictability closer to those of traditional tools.

Admittedly, to finely shape video games according to designer intent, we do not need PCG tools. Game designers and programmers are used to manually crafting video games according to their intentions, using a wide-array of tools (some of which already have PCG techniques incorporated in them). But there is an opportunity in the fact that mediating player experience is still a challenge for game designers. So PCG might be converted into a tool that assists game-designers in fine-crafting player experience, as opposed to fine-crafting games. The question then becomes, what form should a tool that allows fine-crafting player-experience take?

5.1.2 A Human-Centric Approach to Procedural Generation

The Authorial Game Evolution tool (AGE) allows designers to improve an existing base-game until it fulfils an (authorial) agenda for player experience. It is intended as a general-purpose tool, theoretically applicable to a vast array of game genres, design problems, and contexts of use. AGE also has a dual nature: it is both tool and approach, as the tool's functionality assume the adoption of a novel game design process. The overall working of this new approach is in figure 5.2. The gist of how it works is simple: designers create a base-game,

the Archetype, they wish to develop upon. They integrate it with a procedural algorithm that is equipped to vary some part of the base-game's design, the game's Features. Finally, they define a set of target experience indicator values, which will be used as a reference when evaluating generated content.

Once the 3 designer-defined elements are forwarded and integrated into a platform, its actual functioning begins – that is that A-cycle in figure 5.2. At that point, the platform runs without need of intervention from designers and it is at this stage that actual procedural generation occurs. First, there is a phase of candidate generation, where new sets of parameter values are found by means of a search algorithm. Secondly, by providing new Feature values into the Archetype, new candidate artefacts become ready for playing, at which point players play these alternatives.

From play-testing with these candidates, experience data is extracted in the form of experience indicators. Experience Indicators are meaningful variables that track a player experience quality; for example: an average fun indicator can be extracted from player self-reports; and a difficulty indicator could be calculated based on game-play metrics of the number of player deaths. These indicators are compiled and compared with designers' target values, by means of a distance metric. Finally, if the best candidate's quality is not sufficiently close to the values, then this micro-cycle repeats with a new generation cycle. Once the A-cycle ends, a candidate that is close to the target indicators, as well as any experience data on it, is forwarded as output to designers, so they can continue their iterative design process, by choosing whether or not the end-result is satisfactory. If it is not, designers can fine-tune either of the elements provided, forwarding a new design specification, in a new iteration of the B-cycle.

To use the AGE approach, a design team must prepare those 3 items – a game Archetype, and a definition of Game Features and Target Experience Indicators. These 3 elements are that which determines the overall specification of the problem they wish to solve in what regards to their game, and therefore the differentiating aspect of this approach. This is, in essence, the definition of the design domain in which the PCG optimization method will operate, and the notion of generated content 'quality' that will drive the search procedure (which will optimize/maximize quality).

Content – Game Archetype and Features

AGE requires designers to first create a base-game prototype – the Archetype in which variations will be based – that will then be evolved by procedural

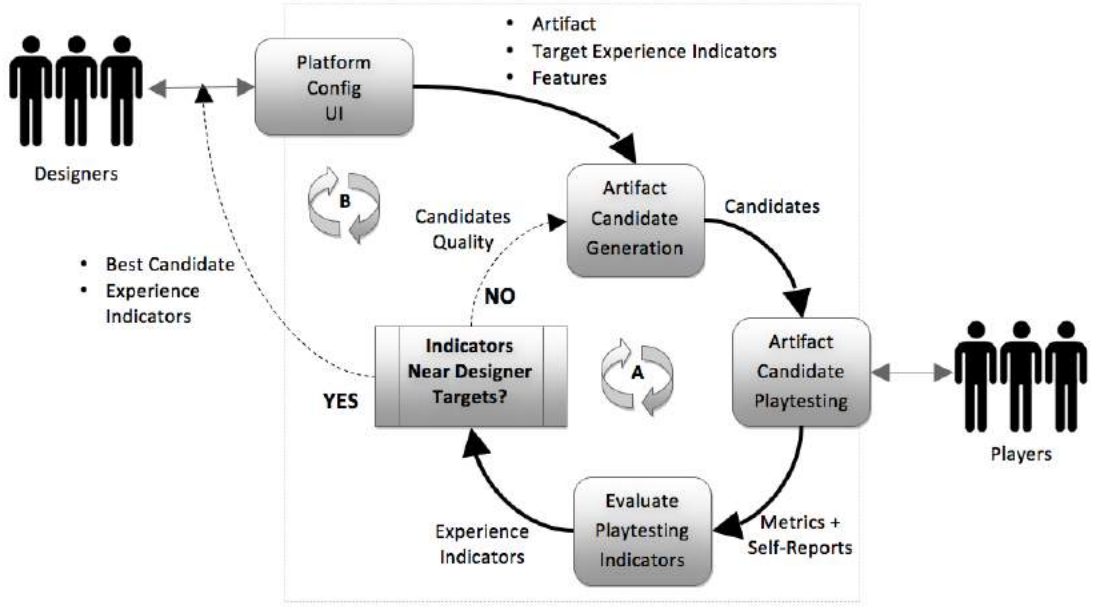

FIGURE 5.2: A step by step view of how AGE was intended to be used by designers.

generation of new values for specific Game Features. In this model, features work as parametric variables that control the game's content.

In terms of content generation, our architecture proposal strives for a minimalist, constrained approach. Since the main objective of the architecture is to provide iterative improvement of artefacts until player experience fulfils specified goals, and not generate totally new video games, there is no need to represent the entirety of a video game, only given aspects of the game that affect desired experience qualities. Here, we enable authors to define which game design variables they are willing to forfeit control of, so that the platform can search for new configurations.

Features represent the set of design elements that designers forfeit direct control of, and which AGE will then vary to define new candidates. They consist simply of a numeric vector with all game parameters designers specify should be optimized; each element in the vector is a one-to-one representation of its respective feature value.

Furthermore, each of these design features must be encompassed as part of a specific domain. As an example, enemies in a level could be defined as a natural number between 1 and 10. AGE will simply generate values from 1 to 10 in so as to find the optimal value for the intended experience. Note it is also up to designers to prepare how the archetype will generate the content for players,

upon receiving the new feature set. There is the assumption here that authors choose these features in a sensible manner, so that they affect the desired target game-play experience elements.

The Archetype is the prototype designers created, deprived of the feature-defined content, i.e. the base-artefact of which all variations will evolve from. In continuing with the designer-centric stance, it is assumed that this Archetype is implemented by designers from the ground up. Naturally, all these elements involve a configuration process carried through a platform user-interface (determining, for example, the types of target indicators, their domain, etc.), and direct integration by means of computational interfaces (methods for publishing and subscribing raw data from the experience indicators).

Content Quality – Designers' Target Experience Indicators

Content quality is measured by an inverse distance metric between designer-defined Target Experience Indicator values and average values of those same Indicators as mediated to players during play-testing by procedurally generated candidate artefacts. Quality assessment is to be solely on a designer defined basis – this ensures a great deal of creative control lies on designers, as it is they that define what constitutes a game's success.

As a simple example, say a designer wants to guarantee a small percentage of player game-overs for a given level. He would then set a target value of 0 for a metrics-based experience indicator that tracked the maximum number of player game-overs per level. Generated levels would then be evaluated in function of how many game-overs players had had in them; a level where no player died would be optimal, with maximum quality, and levels where one or more players died would have a lesser quality.

We can define how quality function works, in formal terms. The purpose of the quality function is to verify if the current game solution's features f provide an experience whose indicators i are in line with designers intentions i_{target}. To maintain quality as a positive and semantically coherent indicator (higher values meaning better quality), a normalized inverse distance (d) metric is used. To guarantee normalization and to afford designers control over how quality is attributed, a maximum indicator distance constant d_{max} is used to define the

boundaries where indicator quality is above zero; see function 5.1 for the details.

$$d_n = \begin{cases} 1 - \frac{|i_{target,n} - i_n|}{d_{max,n}}, & |i_{target,n} - i_n| \leq d_{max,n} \\ 0, & |i_{target,n} - i_n| > d_{max,n} \end{cases} \tag{5.1}$$

This allows designers to fine-model a triangular quality function, centred on the ideal value i_{target} and with base diameter of $2d_{max}$ for each experience indicator, as figure 5.3 illustrates.

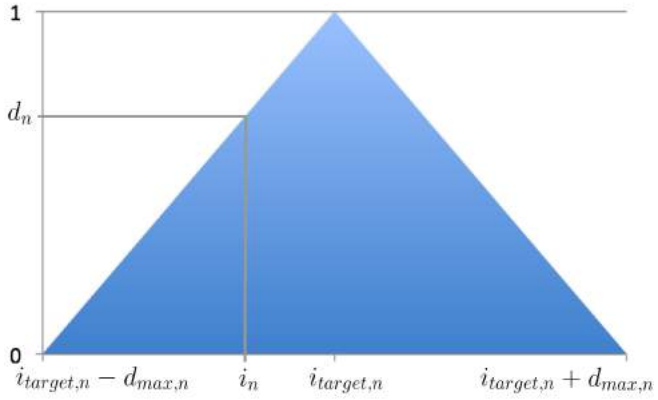

FIGURE 5.3: Visual representation of the distance function used in candidate quality evaluation. The horizontal axis represents experience indicator values, and the vertical axis maps their corresponding distance metric values. As experience indicators come closer to the target value, the distance metric comes closer to its maximum value of 1; conversely, it diminishes as it moves further from the target, becoming zero once it is equal or greater than the maximum distance constant $d_{max,n}$.

Finally, because designers can provide a set of experience indicators N, they can give a specific weight to how each distance metric is considered in the optimization process. Overall quality of a given set of features f is then given as a weighted sum of the distance metric, as seen in equation 5.2.

$$q(f) = \frac{1}{N} \sum_{n=0}^{N} \omega_n d_n \tag{5.2}$$

Unlike EDPCG examples intent on optimizing player experience based on player preference (Yannakakis and Togelius, 2011), this proposal does not intend to let players determine which experience they prefer, nor to allow their subjective feelings to impose as the sole metrics of success of the experience. Rather,

the point in this approach is to let designers determine how player experience is mediated by the artefact, which means that there is a second-order evaluation process involved.

With AGE, designers can model an indicator that measures user satisfaction (for instance, supported by average player reply to a self-report questionnaire covering how much pleasure they derived from a specific content unit), and then try to maximize it... but they can also complement user-satisfaction with other indicators, ignore user-satisfaction altogether, or taking it to the limit, optimize their game to mediate an experience which players find negative, frustrating or uninteresting. Designers are who gets to choose. Users' opinions are therefore completely removed from the quality assessment equation in favour of authorial control – their experience is the basis on which quality is assessed, but it is so according to a mapping that is defined and modelled by game designers.

The idea of this design was to have an open interface which allowed designers to determine the criteria that should be fulfilled in terms of player experience. Through this interface, designers would be able to establish which key indicators were monitored by the system, which target value range should be found by search algorithms, and how each indicator was weighed in terms of the quality function.

Content Generation – Search Engine

Parameter optimization was, at this stage, the most challenging of the architecture's methods, for the choice of a search algorithm required understanding of the search space properties. Since we could not find any systematic study of the relationship between variation in game design features and player behaviour, we needed to acquire some sensibility to this issue through our own research. It was not our aim to study this in such a way as to provide definitive knowledge on this relationship, merely to help answer some fundamental questions for our approach.

Is the search space smooth and differentiable? How much time and how many players are needed for player behaviour data to converge into a meaningful estimate of quality? Is parameter variation impact in experience predictable? Data that helps answer these and other questions needs to be obtained before such a critical decision as choosing a search algorithm for optimization. A first preliminary test intent on tackling these issues is described in section 5.2.

Hypothethical Example of AGE's Design Process

Common scenario: an action game has been in development for quite some time and its levels need balancing. Today, each of these would require extensive monitoring of play-testing sessions, probably using some combination of game-play metrics, questionnaires and interviews. Data from these sessions would then be processed, digested and analysed by game designers in search of issues that needed solving – for example, a high number of per-player deaths in a specific level. That level would then be redesigned with that issue in mind, say by changing the number of opponents on that level.

With AGE, the same scenario would occur in a different form. Once the prototype would be ready, it would be connected to an AGE platform, yielding access to experience data. An interface would allow the establishment of target experience indicators; in this case, designers would want to control an indicator that measured the number of player deaths. They could select a specific range of acceptable values for a game-play metric 'player death', say a target value of 3 plus minus 1, and assign a set of design features to be varied in order to achieve that target, such as the aforementioned number of opponents.

The system would then run play-testing sessions with games that have varying number of opponents, and in each round of tests would assess the proximity (hence, the quality) of the measured metrics to the target values; should data not be compliant with the desired distribution, it would continue to search for better numbers of opponents, guided by the search algorithm. Once goals are achieved, the optimization process ends. Designers would then, if needed, manually look over experience data and verify the end output and the found feature set.

Taxonomical Classification

According to our own Taxonomy (see chapter 4, AGE provides designers greater agency and control over generation procedures by affording an Explicit mode of Evaluation – Quality Definition by way of defining Target Experience Indicators –, as well as generation Configuration (particularly Content Parametrization via definition of Features and their coupling to generation procedures), and the forwarding of Base Design elements which are used by the PCG system (The Game Archetype).

Additionally, the AGE approach is open to Editorial Control in terms of implicit Evaluation (as it is a design-time tool and authors are given the best

candidate), as well as Method Selection and Method Parametrization in terms of Generation (as we aimed to design a UI for search algorithm parametrization), though these are not a core part of the system's design.

In terms of the Computational actor, it has no impact in the Evaluation process (as it only applies a formula explicitly dictated by Humans); and in the Generation task it uses an Optimization Strategy during Design Time. Being a general purpose tool, in theory it can be used for any Content Type that can be defined according to our Features metaphor. The diminished role of the Computational Actor should be noted, as this is by design. By bringing human designers into the forefront, we are placing more responsibility in them, and less in the hands of algorithms. The consequence of this choice is that end-results are all the more dependent on human creators' vision, creativity and capability to use AGE.

Maintaining the focus on human designers, players have no impact in terms of generation of content. Their only role is providing experience data that is used used in evaluation procedures (Experience Model Based Evaluation), strictly during Design-Time. In AGE, the Experience Model is defined by designers through Target Experience Indicators.

5.1.3 The Value Proposition of AGE

The AGE approach is a materialization of an experience-driven design paradigm (Hassenzahl, 2011), as it shifts designers work from establishing artefact qualities, to the design of its experiential qualities. Therefore, despite having some minor degree of autonomy – as PCG evolves certain content aspects of the base-game –, this autonomy is constrained both by supervision of the generative process (through definition of the search-space), and especially by his explicit control of the intended player-experience. The end-result, from a material perspective, is somewhat unpredictable to the designer (because he only controls how the search will occur, and not the specific set of material qualities of the end-result), but the experiential qualities will be in line with his/her desire, therefore increasing predictability and control over the end-experience in the designer perspective.

Because the system is – at least theoretically – problem-independent, it allows designers to apply this approach to any design problem they see fit. It provides no a-priori barriers to appropriation, and its ideological stance – to be author-centric – is thus upheld.

Viewing this from the perspective of the Mechanics Dynamics Aesthetics framework (Hunicke, Leblanc, and Zubek, 2004), what designers are asked to do is propose a generation space for certain game Mechanics, while providing a set of quality measurements that can validate Dynamics and/or Aesthetics according to their creative goals. The game design process proposed by Fullerton, Swain, and Hoffman (2008) is analogous to this process, but it is the designer's role to experiment (meaning: vary and evaluate) different mechanics. What AGE does, in a nutshell, is to automate and take (a part of) the burden required in exploring that design-space until the desired player experience is achieved.

Note that we do not claim designers could or should use AGE to solve all their game design-problems; depending on their experience and agenda, many player experience qualities might be more easily achieved with traditional, non-automated exploratory design processes. Indeed, AGE seems potentially useful for cases in which designers wish to achieve a certain experiential effect but do not know the exact set of material qualities the game needs towards effecting that end. This may include cases of inexperienced game designers, designers trying out creation of games in genres they are unacquainted with, or simply experimental prototyping where the end result is unknown.

We can foresee a number of potentially valuable contexts of use for AGE. In early prototyping phases, it could be utilized so as to facilitate generation of novel artefacts with unusual feature combinations – just as long as there is a defined set of experiential qualities that designers seek. In production phases, it could be used in content production – of levels, worlds, missions – by taking an existing archetype and then varying its features until certain experiential typologies are mediated to players, e.g. a level with a certain difficulty, a world that takes a specific time-span to explore, a mission that imposes a given play rhythm, etc. Finally, AGE could be used in later stages of production in balancing and fine tuning of any game design elements, e.g. character/faction/item/weapon preference in multi-player games.

Like all tools, using AGE comes at a cost. Learning how to utilize the AGE tool to achieve a given effect – getting to grips with its mode of working and becoming proficient with its search algorithm design logic (knowing which features to evolve, and experience indicators to target, etc.), is likely to require significant learning, as it effectively revolutionizes part of the game design process. AGE is, in a sense, a meritocratic approach. It does not solve problems creators themselves cannot solve; instead it shifts responsibility back into their

hands. The quality fo the end-product of an AGE-powered design process depends, in great measure, in creators' skill, as both authors and users of the tool.

Furthermore, creating an archetype and incorporating it with AGE and content generation algorithms that take procedurally generated features requires non-negligible programming effort. Comparing this to traditional processes, this additional cost is balanced by AGE then not requiring designers to manually author revisions to their game and its content when experience evaluation procedures find prototypes not fit for their aims. Whether AGE's value proposal off-sets these costs is, of course, dependent mostly on designers and their appropriation of this approach.

5.2 Game Features Variation Experiment

In the first experimental procedure of this research, we wanted to gather empirical data that could serve to support one key aspect of AGE's design. At this moment, we were still not sure how to search for a good set of features, and required better understanding of how variations work and which effects they elicited. Thus, one of key questions was what sort of search-algorithm to use in terms of optimizing the game features. At that point in time, another research project had required the development of a shooter game named Recycle (Moura, 2011) that included a game-play metrics system used to track players movements and actions. Given this opportunity to test a non-PCG prototype, we devised an experiment meant to determine if we could use AGE with a simple, configurable search algorithm, that worked analogously to a PID-type thermostat.

Our idea was to help designers control the feature generation process, and control it as much as possible, so that the end-effect would be in line with their authorial intent. Thus, we hypothesized that there was a direct foreseeable relationship between specific feature variations and experience indicators. If that would be the case, then designers could attribute feature evolution a direction for the feature search algorithm; for example, if the difficulty experience indicator would be higher than a given threshold value, then a game feature that encoded number of obstacles and enemies should decrease, and vice-versa.

As a secondary goal, we wanted to gain some experience on how a play-testing session with varying game features might occur. Traditionally, play-testing sessions have players experience one given game-object, but in the AGE approach, play-testers are exposed to a number of fluctuating game-objects. We wanted to obtain data on how players would understand the boundaries that separated each game candidate (and its implied set of game-features), as that is a needed condition for player experience data to be meaningful. Potential for pollution of player experience data exists – as experiencing a given feature set might impact later experiencing of other feature sets.

5.2.1 Experimental Setup

Let us assume a hypothetical author, whose intent is to create a multi-player first person shooter game where tension is built by allowing players to both cooperate and compete in the same space. However, to make the game more interesting and dynamic, and preserve some amount of player choice, he does not wish, as would be the simple answer, for the game to merely force players to act in a given way, by statically assigning players to teams as is common in on-line shooter games. His purpose is for the game to convey, successfully, that players should actively band and disband teams as they see fit for their objectives, as it better reflects his ideal of how the video game should be played and also how it can become more entertaining and rich in terms of player experience.

This hypothetical author creates a multi-player shooter where players compete for points, but are allowed to ask other players to join them in a team, as well as being able to, at any time, betray their team and go solo once more. The author also wants this pattern of behaviour to be as dynamic and frantic as possible. Assuming all else remains equal, can an author induce his target player behaviour by simply altering one game design feature?

The base game design consists of a standard multi-player deathmatch, where players can shoot each other's avatars in order to gain points to win. Figure 5.4 shows a screen of the game. Besides killing, players can dynamically forge alliances with their opponents – to do this, they must simply query another player and receive a positive reply. Players exit teams whenever they betray one of its members. While solo, players gain points by shooting and killing opponents and filling their individual score, whereas in a team, they accumulate points for both their personal and collective score. Personal score reflects only an individual player's actions and therefore subsists even after leaving a team,

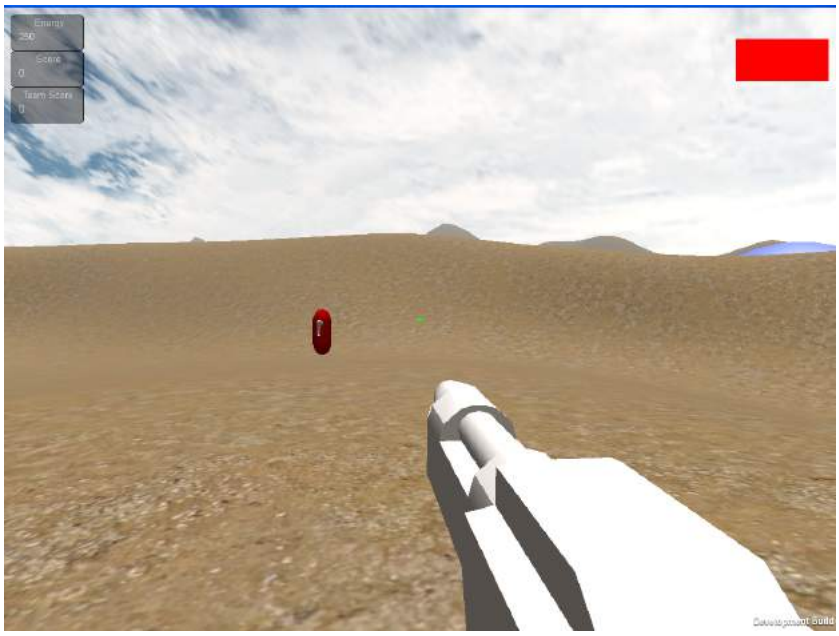

FIGURE 5.4: Screenshot of the videogame used for testing. Current score as well as any score accrual are directly communicated to players, as well as the action which originated it. The coloured square indicates membership to the red team (white would symbolize neutrality or solo play).

while the team score is an accumulation of all team players actions, and thus is lost once a player leaves his respective team. Rounds are played in succession, with each game round presenting a target score as a goal, and with a ranking screen shown to all players in the end, distinguishing who won what points and whether the winner was a team or not. Whether the goal is achieved through players' personal score while playing alone or their team score while banded with other players, is irrelevant to the game.

5.2.2 Methodology

With a game-theory frame of mind, we created 3 different feature configurations that change how players gain points for their personal and team score, benefiting specific behaviours and penalizing others. The idea was that by augmenting team benefits while minimizing individual ones, players should gravitate towards team-play and therefore, their experience should display behaviour coinciding with team-play. If such is the case, game-play metrics-based player experience indicators referring to number of team proposals and team accepts should increase. We carried out two different trials to verify this hypothesis with each trial varying a single, but different game design feature.

For the first trial we changed how score was distributed when players were in a team. In the first session, score was attributed equally to both player and team; in the second session, whilst in a team, players would receive only a fraction of their accrued score C (equal to the team score, divided by the number of players), whilst the team would receive it fully; in the third session, whilst players were in a team, personal scores would not increase. The logic for this variation was to induce players to play as a team, with each session providing even more incentives for players to belong to a team, and not leaving afterwards (as their personal score would pale in comparison).

During the first trial, players voiced complaints of lack of fairness in the distribution of points and a lack of personal distinction face their team-peers, and thus we tried a different set-up on the second trial with a similar intent, while hoping for the same outcome. Thus, to guarantee fairness, players in teams received the same points for their personal and team score (always the base score divided by the number of players in a given team). What varies is a multiplier that caps both scores when players are in a team, varying from 0.5 to 1 to 2 along the three sessions. Feature variations are detailed in figure 5.5.

In both trials, feature variation was done asymptotically, based on the idea that as long as players experience variations in the game design in succession, they should witness these in such a way that does not provoke such a disruption in the game to the point of making it disagreeable. Consequently, if our hypothesis was verified, the number of team proposals and accepts should have increased asymptotically, whereas random variation would denote a rejection of the hypothesis.

Besides a description on how to play the game, players were given an explanation of the context of the experiment – how its intent was to measure the impact of different game features in play styles – and were warned at which point they would experience a novel game design variation, though no specifics on the actual design were ever communicated.

Tests were carried out in a laboratory; to motivate potential participants, a small monetary compensation was given to participants. 10 male subjects responded to the 1st trial call, ages comprised from 23 to 40, and 6 male and 1 female subjects responded to the second trial, ages ranging from 24 to 27. The small number and skewed distribution of subjects is surely not ideal for matters of statistical significance, but we think still serves to give valuable data. Note, of course, that since this was an exploratory study with a small sample, we can

1st Trial			
/	Setup 1	Setup 2	Setup 3
Ind. Score	C	C	C
Ind. Score (in team)	C	$\frac{C}{n_{team}}$	0
Team Score	C	C	C
2nd Trial			
/	Setup 1	Setup 2	Setup 3
Ind. Score	C	C	C
Ind. Score (in team)	$0.5 * \frac{C}{n_{team}}$	$\frac{C}{n_{team}}$	$2 * \frac{C}{n_{team}}$
Team Score	$0.5 * \frac{C}{n_{team}}$	$\frac{C}{n_{team}}$	$2 * \frac{C}{n_{team}}$

FIGURE 5.5: Feature values for the number of points attributed for a given action, for a player playing solo (Ind. Score), for a player while playing in a team (Ind. Score in team), and for the team score as a whole. The table shows how the set-ups were changed from session to session in the 2 trials. C stands for a constant awarded for different actions (accurate shooting, opponent kills), and n_{team} is the number of players in a specific team.

only look for indications to base our design on, and not significant evidence that establishes these hypotheses as true or false.

5.2.3 Results

The first trial showed results pointing towards a consistent increase in team formation: set-up 1 was tested for 15 minutes and 45 seconds, the second set-up 15 minutes and 30 seconds and the third for 13 minutes and 36 seconds. The expected effect was verified. Number of actual team creating proposals increased steadily in the 3 different set-ups, as well as the number of accepted team requests (see figure 5.6).

We also measured the total number of team proposals, which includes successful team proposals, as well as redundant or ineffective team proposals directed at players' own team and opposing teams members (the game does not allow merger of different teams) and team proposals that miss any player target – in this indicator as well, we find a steady increase of the average (see figure 5.7).

Now, this effect could be of simple random statistical variation or even a matter of adaptation to the game; as control variables we can look at neutral variables to the team creating process. Figure 5.8 shows the first trials measurements

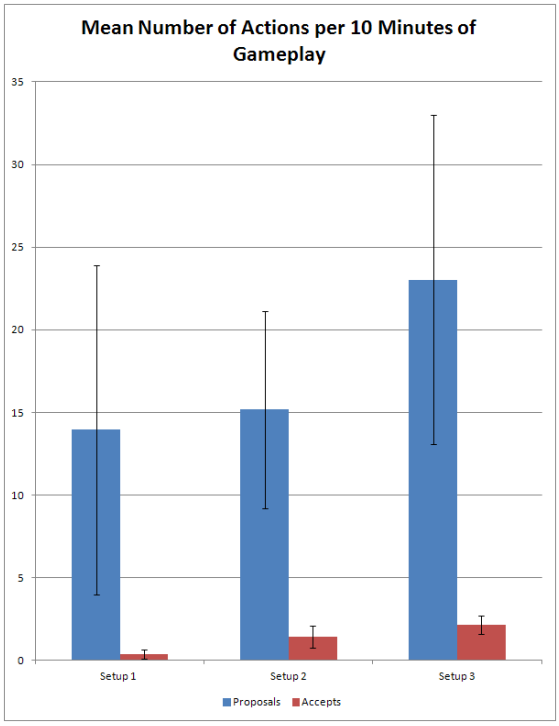

FIGURE 5.6: Trial 1's mean number of team creating requests (Proposals) and accepted team invites (Accepts) per ten minute intervals. Black bars denote standard deviation.

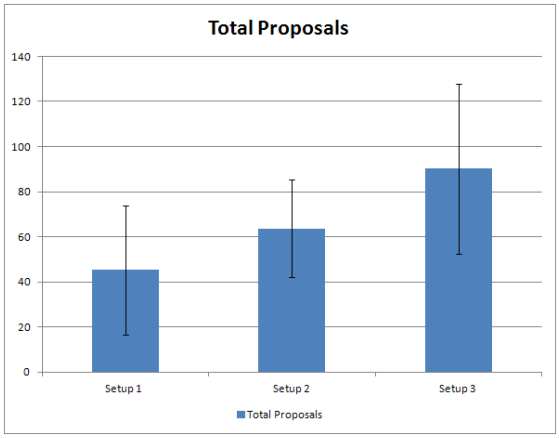

FIGURE 5.7: Trial 1: the chart shows sample's mean number of total team creating requests per ten minutes. This is indicative of the rhythm of team proposals on each set-up. It includes missed requests as well as redundant attempts. Black bars denote standard deviation.

of three neutral variables: shots fired, running actions and jumps. The number of runs and jumps seems unaffected, while shots show a slight decrease, that we speculate arose from increased proficiency on the part of players in playing the game. So, the steady increase in the average number of team-proposing and

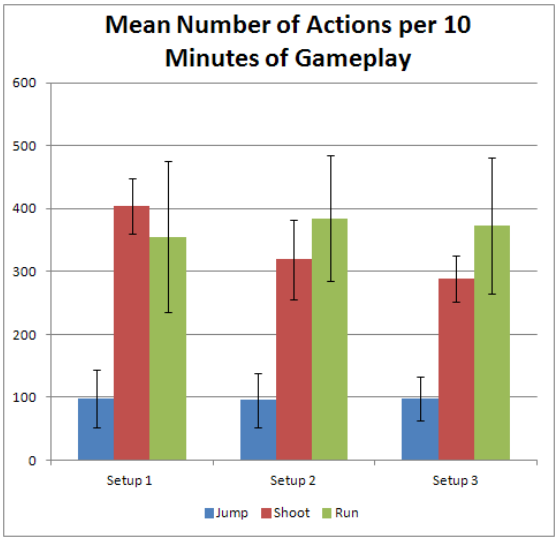

FIGURE 5.8: Mean number of jumps, shots and running actions taken by players per 10 minute period in Trial 1. Black bars denote standard deviation.

team-creating actions is either on account of our communicated variation in terms of game design, or the passing of time.

The second trial ensued for a number of reasons. Firstly, to confirm the pattern with new players, extending the player sample; secondly, to measure if this effect could be extended for increased periods of time (the first session lasted less than an hour); thirdly, to correct minor flaws in the game interface which players complained about; and finally, to address issues of personal value and fairness which some players made note of questioning during the first test.

Feature variation was changed as aforementioned and this time trials lasted longer, even though there were less test subjects. Set-up 1 was played for 44 minutes, 2 for 39 minutes and the final set-up was cut to 28 minutes (one of the subjects had an emergency, and left mid-game). A similar approach to the first trial was tested, only using different feature variations for the same end. The use of a score multiplier that benefited team-play should, in theory, increase the number of team creating actions, by making it more efficient to play in a team.

However, the previous pattern was not verified (see fig. 5.9 and 5.10). Players did not tend to play more in a team as time went on. In fact quite the opposite occurred, as the rate of team proposals and team acceptances diminished steadily from session to session. This effect was puzzling – why would players not tend to create more teams as time went by and take advantage of the improved accrual of points? Data from the first trial indicated that players would follow

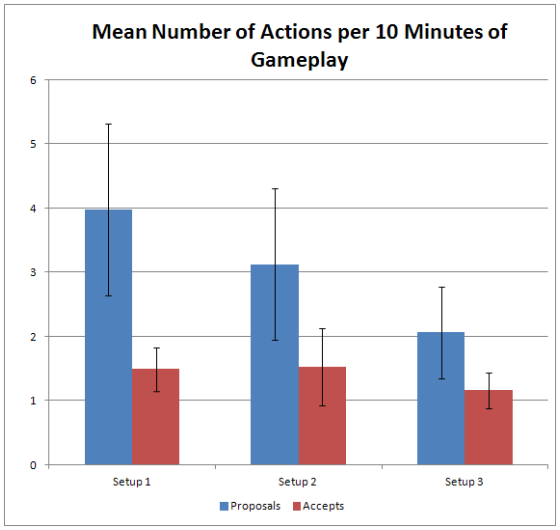

FIGURE 5.9: Trial 2: Mean number of team creating requests (Proposals) and accepted team invites (Accepts) per ten minute intervals. Black bars denote standard deviation.

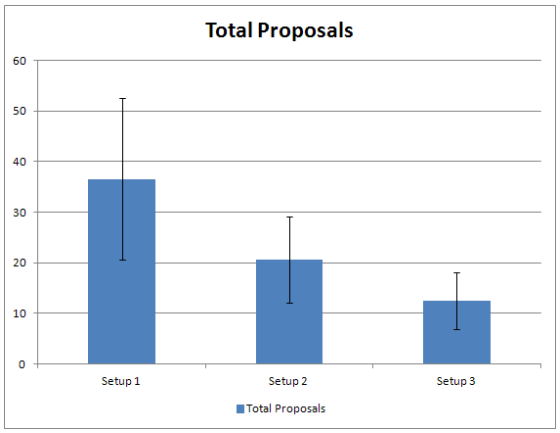

FIGURE 5.10: The chart shows Trial 2's sample's mean number of total team creating requests per ten minutes. It is indicative of the rhythm of team proposals on each set-up. This includes missed requests as well as redundant attempts. Black bars denote standard deviation.

a trend for the creation of more teams; the opposite is true for the second trial, while neutral actions show no such tendency, see figure 5.11.

One attempt at explanation could be that players had become more effective at banding in teams and therefore, given more time, were creating less teams throughout game-play, simply remaining together in the already banded teams. But such is not the case: by looking at the average number of in-team players per second in figure 5.12, one clearly sees that the value is falling. This result further indicates that players diminished their tendency to play cooperatively in

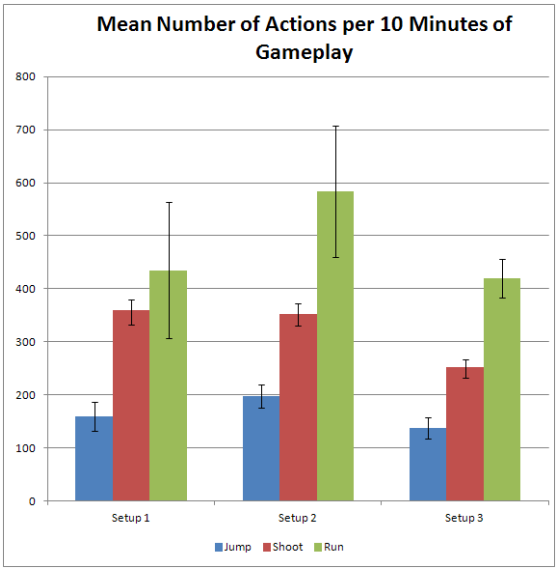

FIGURE 5.11: Trial 2: the chart shows players' mean number of jumps, shots and running actions taken by players per 10 minute period. Black bars denote standard deviation.

the second trial.

5.2.4 Discussion

Trial 1 results clearly showed the expected effect, the opposite being true for trial 2 results, but the reason for these disparate results is not entirely clear. Assuming this is not just a random variation, product of a change in population, what could be the cause? The different feature choice could account for a marginally different player interpretation, yet both set-ups benefit players participation in teams.

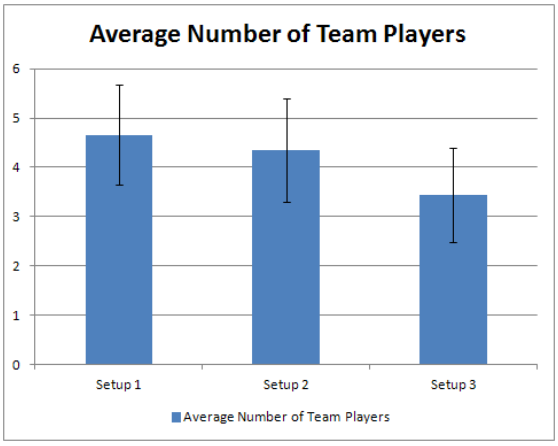

FIGURE 5.12: Trial 2's average number of players in teams per second. Black bars denote standard deviation.

We could also raise the question of perception of relative benefits. The initial trial gives players in a team the same score they would otherwise receive if playing solo, but the advantage of several players contributing to the same score pile is extremely significant. A player outside a team would struggle to compete with 2 or more players if they received exactly the same score as he did whenever he shot an adversary. This notion also led us to change the original feature set-up: players noted the playing field was too generous for being in a team. Players might not have acted on the variation and simply understood that the actual baseline design was already benefiting team-play; increase of the number of team-actions could just signify increased awareness of the fact. This would account for the rise in player teams in the first trial.

In regards to the second trial, the inverse might be true: because players in teams have a balancing penalty (that sees their score being reduced in proportion to the number of players in each team), players might feel that going out solo is the better option, since there is no balancing penalty while scoring solo. This would, of course, presuppose players' disregard for the advantage in having several players contributing to the same score amount. The immediate response that players have when shooting an opponent might be more appealing while playing solo, since their score is higher. It is only when team-mates pitch-in to the score that the advantage is made clear.

If this line of thought is true, then it may be that our proposed variations are lateral to actual player perception of the game-play experience and consequent behaviour and experience. Because information in a game is complex and multi-faceted, and game-play already requires concentration and effort on the part of players, they may naturally disregard finer aspects of the game's logic, even should it be communicated to them that game rules have changed.

Another question that can be raised is that of preconceptions specific player populations might have on how to play a specific genre and desirable behaviours in game-play, the notion of success in that kind of game, and social pressures to perform in a manner aligned with some standard of behaviour. The relatively small size of the populations in these trials, especially in the second case, might also influence what can be the space of possible behavioural outcomes, for instance, by significantly limiting the size, multiplicity and diversity of teams and their social behaviour.

These two trials however, do not present conclusive data to make such assessments. They do present us with interesting questions deserving further

research, hinting at the hypothesis that small changes in game features might not be sufficient to drive player behaviour at all, perhaps not even sufficiently affecting player experience.

5.2.5 Design Revision

As to the overall implication of the results for the AGE design, it was clear that feature variation did not consistently have the expected translation in terms of player behaviour. The first and most important change to AGE's design then, resulted from that understanding that feature variation can have extremely unpredictable effects. While part of the problem may rely on outside causes, this signals a need to avoid search algorithms that assume functional relationships between features and experience indicators. Assuredly, the measured effect might be due to a poor choice of features in respect to the second trial; but we must keep in mind how simple this experiment was, and how direct the relationship between chosen game features and experience indicators was (or at least, seemed to be). In complex cases, the sort of unpredictability will likely rise.

Therefore, we can reject the enunciated plans for a designer guided search algorithm. The idea that we could afford game designer simple control over how feature variation would evolve was as such dropped from the design. Instead, based on this data, it is our opinion that AGE's search algorithm cannot make any sort of a-priori assumptions over how features will impact experience.

Our solution then, was that **AGE will use Genetic Algorithms (GAs) for feature search procedures** in future experiments. GAs were selected for being a family of extremely versatile, sub-optimal, stochastic, general search methods that can be easily adapted into a myriad of different problems (Haupt and Haupt, 2004). What this experiment suggested is that the landscape is unpredictable, problem and population-sample dependent, and non-linear and dynamic over time. Player experience and behaviour cannot be directly dictated by game design features, which means there is a non-deterministic relationship between the two. Different players react to the same rules of the game in different ways, and the same players may react differently to the same rules at different times, etc.

An adaptable search algorithm that uses non-problem specific heuristics and makes few assumptions over the search-landscape seems a good fit for a class

of problems whose search-landscapes are highly variable and dynamic. Unfortunately, the lack of an unknown (as it is problem dependent), well-behaved search-landscape also means that there will be no guarantee of the algorithm's convergence or its finding an optimum (whether local or global). A good selection of features and experience indicators will be left up to designers.

It is important to clarify that the choice of the methods' generation algorithm and its technical specificities is not a focus of this research. A choice of a search algorithm simply had to be made to prototype AGE, and based on existing data, this was our choice. This does not mean that AGE needs to use a genetic algorithm or any other specific search algorithm. First and foremost, this research's aim is to solve how to frame PCG methods in the game design process, in a way that aids designers and empowers authorial design. We are concerned with the design of this tool, and how it can frame the use of PCG algorithms, and not necessarily the algorithm itself. So we accept that the AGE Approach can be carried out by other, more advanced search algorithms and/or other artificial intelligence algorithms. Our literature review showed a wealth of alternatives with proven merits in finding solutions to similar problems as this, and so it seems less of a priority to explore this avenue further when there is so little done in terms of researching how to frame these methods in a design context.

On the other hand, lack of predictability of players' interpretation of signalled changes in the game's design, suggested that communicating design changes to players could help them behave *accordingly*. AGE's game interface could offer better visibility of the aspects of the game's design that vary. However, this would pose a problematic challenge in play-testing sessions – as communicating which aspects vary may induce players to behave unnaturally, i.e. based on player perceptions of how the change should impact their behaviour instead of their unconscious game-play style adaptation to the new game content.

To avoid that effect, we propose that **play-testing procedures should involve communicating what features can change, and when new candidates are being played** – to heighten players awareness of the varying landscape – but not explicitly convey what has changed. To minimize player sample experience indicators' variance, **play-tester samples should be as large and statistically representative of the target population as resources allow**. Additionally, **candidate play sequence should be randomized and players should repeatedly play the same candidate** (to average their indicators, and stabilize their play-style face a new game instance). It is also recommended that player-candidate

attribution should be random and even (so that more or less the same number of players play each game the same number of times). Repeated play-sessions with the same game candidate has the downside of extending play-testing duration (and accordingly, the cost of play-testing procedures with AGE), but it seems a necessary cost so as to make results minimally stable and reliable.

5.3 Proof of Concept – Functional Prototype

Once the functional design specification for AGE was finished, we decided to do a first trial to evaluate if AGE could be used on a hypothetical design problem. The main goal for this experiment was to test the feasibility of this approach. We used the guidelines from the design revision of the previous test (see section 5.2.5), including our choice for a Genetic Algorithm to generate features, and Game-play Metrics as the sole data-source used for player experience indicator calculation, and used this data to implement a first prototype of the AGE tool.

5.3.1 Experimental Setup

The idea of this first trial was to take the 'Infinite Mario Bros' level generator. and impose a significant variation on the original 'Super Mario Bros.' design. The choice for this specific prototype came down to two factors. The first was availability; "Infinite Super Mario bros" is a functional, open-source copy of "Super Mario World" with an included procedural level generator. This means it was a complete game, with well designed mechanics and systems which was accessible to use, and easy to manipulate. The other is that it was a game used in other PCG studies (such as (Pedersen, Togelius, and Yannakakis, 2009)), and furthermore, platform games level generators are some of the most explored case studies in PCG; this allowed us to explore our own alternative in a common ground with other researchers.

Imagine a hypothetical author that wanted to make a *hyped-up* version of Mario, with high-speed and high-stakes game-play, where player experience would be akin to that of doing a speed-run trial. So, this designer changed the core-physics of Mario accordingly: Mario jumps double the height, can shoot more fireballs, and his default state is Super Mario (though he is still turned to normal as he is hurt, as in the original game).

These subtle changes in the game's design affect the experience – the game

Indicator	T. Value	Domain	Sucess Range
Actions	2	$0-6$	1–3
Tries	1	$0-5$	0–2
Time	40	$0-120$	20–60

FIGURE 5.13: The three gameplay indicators that the algorithm meant to optimize, their target value, and the domain that was considered. Also, the minimum condition for success for this trial – to have the best candidate's indicator values nested in the success range.

becomes a caricature of the original, both more frantic, exaggerated and difficult to play. The core game-play system is completed, even though the core experience is broken; the altered 'Infinite Mario Bros.' base game thus serves as this trial's Archetype. The designer wants users to play as he envisioned: very fast, with a high rhythm of actions, all whilst maintaining it playable enough so players can conquer levels without having to retry levels repeatedly.

Now he must define target indicator values for gameplay metrics: **Number of Tries** for conquering a level, as a measure for player difficulty and challenge; number of player triggered **Actions per Second**, as a measure of the experience's rhythm; and level play-through **Time**, as a measure of play-speed. He establishes a target mean value for each of these metrics, as well as suitable maximum range value that defines acceptable variation for these metrics; see table 5.13 for the tested values. That is his experience ideal, materialized into target experience indicators for game-play behaviour.

As a measure of success for this first trial, it was established beforehand that the system would be deemed feasible if it was shown to be capable of generating a best candidate that, at least, could statistically guarantee indicator values between the target and a maximum of a quarter of the accepted range for a population of players (see table 5.13).

Eight features with a varied range of design objectives were chosen to be permitted to be altered by the platform: game speed (24–48 cycles per second), number of cannons (0–5), number of goomba enemies (0–30), number of koopa enemies (0–30), number of holes (0–7), number of item-blocks (0–15), number of coins (0-100) and number of hills (0–5). Apart from speed, all features were given a range from 0 to as high a number as possible, keeping in mind that these values had to produce levels that could not break the constraint of the level size of 480 tiles. This is why the number of holes and cannons (structural level

properties that occupied several tiles), had to be kept relatively short.

This Feature set encompasses one physics feature – Game Speed – which radically alters game experience, as well as impacting game difficulty (as a faster game becomes harder to play). The Speed feature was allowed to fluctuate between the games' normal speed of 24 cycles per second (which resembles Super Mario's original speed) to double that, 48 cycles, which is so fast it borders unplayability. The speed variable is bound to affect all 3 of the chosen target indicators, as it makes both faster and more difficult to play.

Four features were chosen that determine adversarial forces to hinder player progress (cannons, goombas, koopas and holes). These are expected to affect all three indicators, as more enemies and holes impact difficulty, as well as delay player traversal of the level, while increasing opportunities for actions (jump, stomp and shoot).

Two features were chosen that allow exploration rewards, coins and item power-ups. Coins only affect rhythm and traversal time, as collecting them increases the number of actions, while taking time. Coins do not have impact on difficulty, as they do not improve chances of conquering a level, since in this prototype the player is given 999 lives to play the game, there is no tangible benefit to collecting 100 of them so as to gain lives. Item-power ups should provide the reverse effect on indicator impact: mushrooms and flowers ease player difficulty in clearing a level, while only marginally affecting action rhythm and traversal speed (as they are few and dispersed in a level).

Finally, the number of hills feature was expected to be quasi-neutral to all indicators, as it is a structural feat that provides a mostly cosmetic effect (hills are just platforms where player and enemies can go to). It is possible that it could slight increase the number of actions (as it provides a jump affordance) or that it could hinder progress (as it could present a slightly more complex level structure for players to overcome).

5.3.2 Search Algorithm Implementation

Candidates' feature sets are evolved by means of a simple Genetic Algorithm. Each individual is represented by a chromosome that is a vector of real values, each gene corresponding to a given game feature that can be varied. For each generation, the population size was of 12 individuals, plus one extra slot reserved for elitist maintenance of the best individual.

Uniform crossover was used as a recombination method, with a rate of 90%. Mutation operated by replacing a gene to a new random value, and had a starting probability of 5%, and that was increased 0.5% in each generation (these seem large values, but one must consider that for this test, there were only 8 genes and 12 individuals, so lower values would practically exclude mutation from occurring).

Selection was made by tournament for pairs of chromosomes. Evaluation of each candidate's fitness is as follows: in each evaluation phase, all the chromosome population is converted into full sets of design specifications that are stored in the server. Players use a client application which automatically downloads the next available set, which users then play and experience, after which the metrics data from their play session is uploaded to the server. It is from the metrics-based indicator data that candidate fitness is calculated, using formulas 5.1 and 5.2, and then applying them to the problem at hand; see equations 5.3 and 5.4 for details.

$$d_n = \begin{cases} 1 - \frac{|i_{target,n} - avg(i)_n|}{d_{max,n}}, & |i_{target,n} - avg(i)_n| \leq d_{max,n} \\ 0, & |i_{target,n} - avg(i)_n| > d_{max,n} \end{cases} \tag{5.3}$$

$tgt(i)$ the target value for a given indicator i, $avg(i)_n$ is the average value for an indicator i measured from the available sample of game-play sessions, and max_n is a value determined by designers on what the maximum variation they are accepting in terms of the end sample, before the usefulness of the candidate becomes 0. We used the average indicator value of all play-sessions as the indicator value used in the distance function, $avg(i)_n$, and attributed no discriminating weight factor to indicators (so $w_n = 1$ for all indicators), therefore establishing they are all of the same importance to this hypothetical designers' agenda.

$$fitness(candidate) = \frac{1}{3} \sum_{n=0}^{3} d_n \tag{5.4}$$

The evaluation phase goes on until every population member has had at least 5 evaluations, i.e. 5 different play-throughs of the same level (this was the minimum we established that was deemed feasible given the available time-span and number of play-testers). Once this condition is fulfilled, the algorithm will run its evolutionary mechanisms, and a new generation will be evaluated. In

Generation	0	1	2	3	4	5	6	7*
Sessions	76	176	138	107	105	106	184	26*

FIGURE 5.14: The table shows the total number of gameplay sessions evaluated per generation. Generation 7 was incomplete, and as such its data will not be further contemplated (with the exception of new evaluations of the best individual that happened in that final stage).

between, if at any moment the best chromosome produces a candidate whose fitness is above 0.9, an external validation phase is carried out, where it is played at least 15 times, to validate its fitness level with greater levels of credibility. Only after this process does the genetic algorithm resume.

Candidates in the form of feature sets are forwarded to client-applications sequentially, which means that if a player keeps playing consecutively, he will likely experience a given generations' entire population in sequence. However, the system is connected asynchronously, with no control over who, how and when experiments a given set, so play-testers were free to access the game at their own leisure, throughout the one week span of the test.

5.3.3 Experimental Setup

The play-tester group was comprised of 25 Informatics Engineering and Multimedia and Design Master students, 21 male, 4 female, ages 22–26. They were asked to play the game as means of collecting data for research purposes. No incentives were given, and no questionnaires were collected from the population. There were a total of 696 game-play sessions played on subjects' own time and convenience, giving 27.84 game-play sessions per player on average.

Data from the sessions was used for 6 genetic algorithm generations, plus a few more evaluations that occurred afterwards that were not enough to complete a generation's cycle. In total, 68 different procedurally generated candidates were tested until the sixth generation. This means an average of 9.85 gameplay sessions played per generated candidate ($stdev = 5.52$), distributed unevenly throughout generations and play testers (see figure 5.14).

FIGURE 5.15: Evolution of average population fitness, its standard deviation and the fitness of the best individual in the population. Note there is folding in the y-axis.

5.3.4 Results

As can be seen from figure 5.15, the algorithm was able to improve solutions relatively steadily until the final generation, assuring us of its overall convergence. Both the population's average fitness, as well as the best individual's fitness increased moderately until the final generation. Fitness variance in the population shows a tendency for decreasing, though it increases slightly from 4th generation onwards.

While the best results seemed close to ideal, several things are in need of improvement. For one, in this test, the initial setup seems overly positive, with average fitness levels already in 0.8 levels (on a [0.0–1.0] scale). To a certain extent this is to be expected, given that the platform's aim is to iterate on existing designs, and not necessarily to generate an entirely new design from a completely broken one. That being said, the improvement that was measured, while substantial in terms of player experience, did not allow a strong validation of this method's power to achieve the desired target indicators.

Also to take into account, when interpreting the graph, is that the fitness landscape is, in essence, dynamic, as it depends on players behaviour patterns during game sessions. This happens, for instance, when a chromosome passes from one generation to another, and is then evaluated again, which can cause significant fluctuations to its fitness level. One particular instance of this dynamic component affecting fitness levels, is easily perceived in the second generation, where a higher number of gameplay sessions seems to have led to a drift in

values for all individuals, with a significant decrease in both average and best fitness (see chart 5.15 and table 5.14 side by side). This means that part of that initially excessive positive assessment of the population was likely to be due to statistical misrepresentation of the sample. Even so, overall, the test presented a good trend for improvement of solutions, though from these results, it seems it presented a low difficulty problem.

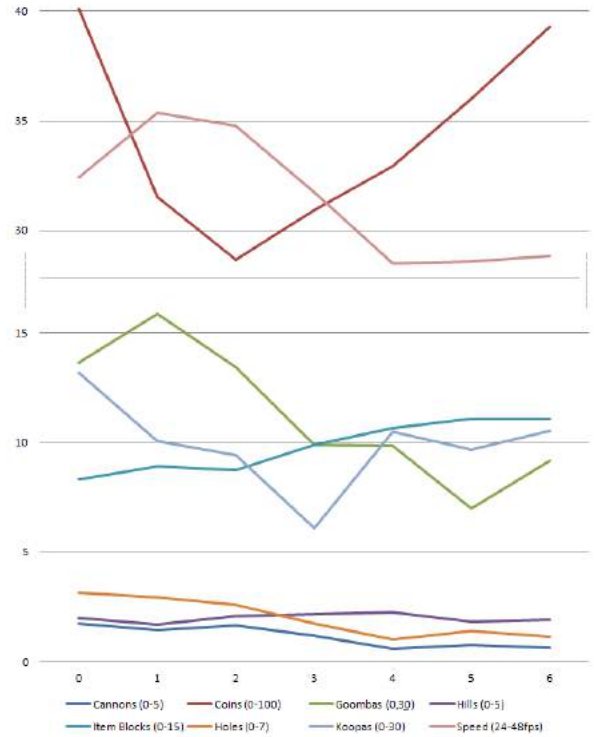

FIGURE 5.16: Evolution of average feature values throughout generations. Note that there is a folding of the y-axis, between 15–30, for ease of visualization.

Figure 5.16 shows how features varied, on average, across generations. Given the tendency for genetic algorithm to improve the average fitness of the population (and therefore converge with target metrics), the major tendencies for features variation mirror an improvement of the candidates. The seemingly more meaningful trends in these feature variations are:

1. a steady value for the number of hills,
2. a slow decrease in cannons, holes, goombas and game speed,
3. a subtle increase in item blocks,
4. an initial sharp decrease in number of coins followed by sharp increase.

The lack of variation in number of hills is unsurprising, given how little they impact gameplay other then providing players with a few extra platforms for

jumping. These could have improved the number of actions, but in this test the effect proved negligible. The decrease in cannons, holes, goombas and game speed, hand in hand with the small increment to item blocks, can be seen as the algorithm seeking to decrease the game's difficulty and its action rhythm, both of which were initially higher than target levels (this trend is visible when comparing this graph with figure 5.17 and target gameplay indicators absolute values). The initial decrease of coins followed by increase around generation 3, are probably the result of trying to adjust the number of actions, as these are above target average before generation 3, and below afterwards.

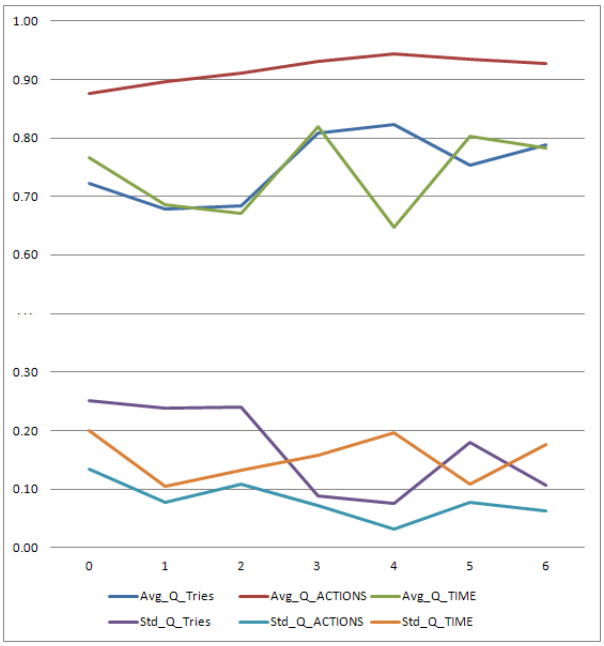

FIGURE 5.17: Evolution of the three quality components, each for one of the indicators ('Tries', 'Actions' and 'Time') used to calculate the fitness function. There is folding in the y-axis between 0.3 and 0.6.

Looking at each of the quality vectors in the fitness variable in figure 5.17, one can discriminate how the algorithm fared at improving each of them. While 'Actions' and 'Tries' seem to follow a somewhat stable convergence curve, tentatively improving on average, 'Traversal Time' seems to oscillate with no clear tendency to improve on a population basis. The reason for this result only becomes clear once the relationship between features and indicators is analysed further, per figure 5.18.

The correlation table establishes clearly why the "Time to Traverse Level" indicator could not be radically improved across generations – none of the chosen game features seem to have a statistically significant relation with its

Correlations

		Cannons (0-5)	Coins (0-100)	Hills (0-5)	ItemBlocks (0-15)	Holes (0-7)	Koopas (0-30)	Speed (24-48fps)
Actions per Second	Pearson Correlation	**-.191**	**.319**	**-.164**		**-.172**	.059	**.333**
	Sig. (2-tailed)	.000	.000	.000	.001	.000	.117	.000
Time Traverse Level	Pearson Correlation	-.015	-.019	-.001	-.015	.049	-.016	-.077
	Sig. (2-tailed)	.694	.613	.988	.693	.195	.675	.042
Tries	Pearson Correlation	**.257**	**-.102**	-.071	**-.199**	**.420**	.043	**.197**
	Sig. (2-tailed)	.000	.007	.058	.000	.000	.254	.000

FIGURE 5.18: Table showing statistical correlation between features and target indicators. Significant values are highlighted in red and greater font size means greater variance prediction.

variation. This indicates that the feature set was insufficient for an adequate optimization of that particular indicator.

This failure of one of the indicators to converge in terms of the overall population, hints at the dangers of a poor choice of 'Features' in finding an adequate solution, as well as the need for subsequent iterations of the cycle, by means of designer intervention (as introduced in the AGE's design).

As to the other two indicators, it is clear that the set of 'Features' was sufficiently capable of producing some degree of optimization. All features with exception of number of 'Koopas' impacted the 'Actions Per Second' in a statistically significant manner, though only three could explain more than 0.3 of its variability: Goombas, Coins and Game Speed, three features that we previously saw were probably varied to improve the number of actions. Number of 'Holes' and 'Cannons' affect negatively the action rhythm, though this is most likely only due to the increase in difficulty – more challenge means a more cautious game-play style which then means slower actions' rhythm.

As for the 'Tries' indicator, difficulty seems to have been mostly influenced by number of Holes, and on a smaller level, by Game Speed and number of Cannons, and as expected, counter-weighed by number of item-blocks.

5.3.5 Best Solution

The best Individual had a final fitness value of 0.96, calculated out of a total of 15 evaluations. Level retries: $m = 1.13, std = 1.85$; actions per second: $m = 2.14, std = 0.63$; time to complete level: $m = 45.26, std = 15.83$. Its feature

set was: 0 Cannons, 30 Coins, 15 Goombas, 1 Hill, 4 item blocks, 2 Holes, 9 Koopas and speed of 28 cycles per second. However, despite the closeness of indicator results to their intended values, the distributions are uneven, thus failing to provide a strong validation for the feature set being able to generalize this behavioural pattern.

To study the extent of the generalization possibilities for this candidate, one must turn to statistical indicators. The standard error for skewness in a population of this small size is 0.580 and kurtosis is 1.121, so any skewness and kurtosis that is greater than the double of these values is indicative of a non-normal distribution. The 'Number of Tries' distribution of the best candidate has skewness 1.739 and kurtosis 2.431, therefore being both skewed and leptokurtic, i.e. significantly non-normal. Skewness and kurtosis values for Actions Per Second were -0.679 and -0.844, and for Time To Traverse Level were -0.697 and 1.791 respectively, therefore not significantly non-normal in both these aspects. Consequently, it is possible to use the Shapiro-Wilk test of normality for both distributions, verifying that it is not possible to reject their normality ($sig = 0.203$ and $sig = 0.478$ respectively).

Given these facts, it can be asserted to be somewhat likely that these two indicators are samples of a normal distribution. With this in mind, and assuming the play-tester sample to be representative, it is possible to calculate how close this game was to achieving the target metrics for the population at large. The confidence interval, at 95% confidence levels, for the"Actions Per Second" distribution is $[1.79, 2.48]$; for "Time To Traverse Level", $[36.50, 54.03]$, at 14 degrees of freedom. Both these ranges are well within the conditions of success for this trial, and closely match what was reasonable to obtain from such a procedure – to have a mean for each experience indicator that is close to designers' wishes. And while it is not possible to extend calculations for the 'Tries' distribution, at least its average did not veer off course (see figure 5.13 for success ranges). Admittedly, the pre-established values for success turned out to be underwhelming considering the results, in line with the higher than expected fitness levels for candidates. In any case, if one sought to improve things further, then more evaluations would be needed; at such a point, it would be up for designers to decide whether the trade-off is worth it.

To conclude, of the three target indicators, the best candidate achieved appropriate mean levels for all, though only 2 of these produced sound distributions that hint at generalization of this success. Only these 2 can, with an adequate

confidence level, be seen as statistically trust-worthy representatives of the candidate's indicators imprint. So, this method was a success for the two indicators that the Features were capable of provoking variations (as statistical correlations attested to).

5.3.6 Discussion

Given the obtained results, it is with confidence one can conclude that in this instance, it was possible to develop and implement a simple algorithm that subscribes to the proposed Author-centric approach to PCG. It proved capable of achieving sub-optimal values for metrics-based experience indicators, though only as long as the design space was well defined (meaning, when features were shown to be capable of significantly affecting said indicators). The fact that it converged, for this specific case, is a key give away. This entails that the experimental trial achieved its goal: to find sufficient evidence to sustain the feasibility of the AGE approach to design problem solving.

Furthermore, its partial success in terms of which experience indicators could be improved, highlights a key AGE limitation: it can only find good candidate solutions if selected feature variations can impact target experience indicators. This effect can be examined through statistical correlation analysis between the two variables; only if it is present can designs problem be solved. This puts greater weight on designer choice.

Further compounding the issue is the question of problem complexity. This test-case was purposefully kept simple, with less than 10 features to evolve, only 3 target experience indicators, and, as we have seen from the results, a good starting point. This was a comparably easy problem to solve. Designers can end up establishing large search-spaces, rugged fitness landscapes that are difficult to traverse, or simply very rare and elusive quality targets; all these can increase the cost to find a good solution in a reasonable number of evaluations... and this is of course admitting there is a solution to be found in the first place.

The configurable, tool-like nature of AGE means that it cannot guarantee success in the feature optimization process; ergo, it can only solve the design problem if designers model the problem in a way that makes it solvable by an algorithm. Like all tools, greater designer agency demands skill and experience for the proper effect to materialize. This does not mean that the tool is not to be adept at optimization; only that it can only find solutions insofar as the problem is well-defined.

Results also showed that some elements are in need of revising. The simple genetic algorithm that was used for generation purposes did achieve moderate improvement to the game instances, though the fact remained that convergence was achieved from an exceedingly positive starting point. Further, both the high variability and dynamic nature of the fitness landscape impacted significantly its optimization process. To solve this issue, it is recommended that AGE is used with a higher number of minimum game-play sessions (greater than this test's average of 10 play sessions) to evaluate candidates (therefore decreasing the likelihood of more drastic fitness variations), and that the selection process is adapted to account for the varying nature of the search-landscape.

Additionally, the difficulty in controlling the maximum variation of a given experience indicator suggests that greater flexibility could be given to designers in terms of defining the distance metric, and its corresponding quality/fitness function. Future revisions of the AGE design should strive to provides designers greater degrees of control over how much variability they intend to have in their candidates beyond the pre-programmed distance metric used up until this point.

This section concludes the initial, functional design specification of the AGE Tool and its working process. The proposed design and guidelines, when coupled with this feasibility experiment – that shows this tool can solve simple design problems – serves as a *de facto* answer to our initial research questions, specifically, *"How can game designers use experience evaluation methods and procedural content generation algorithms to create prototypes that mediate intended forms of player experience?"*. The answer, of course, is AGE, as defined in this chapter. Results from the Mario Experiment are the proof-of-concept that shows that AGE, at least from a purely functional point of view, is a valid proposal to improve game design processes. What remains to be seen is how it can be used, and if it does indeed improve those processes (further dissection of the research answers lies in the final chapter, particularly section 8.1).

Up until this point, AGE was simply a design and a functional prototype, but it was not a tool as of yet. For it to become so, it needed a way by which designers could use it, and so the next step in this research effort was geared towards designing an interface, and that is the topic of the next chapter.

Chapter 6

AGE's Interface Design

Having a proof of concept that implemented the functional underpinning of AGE, still left out the important aspect of designing how users could interface with it. Towards gaining a better understanding of user needs and clarifying how AGE could shape game design processes, we turned to designers themselves, and employed Participatory Design in the design of the AGE interface. The outcome of this work was a series of findings related to how prospective users envision their interaction with AGE, and an interface model to build and user-study a prototype for such a system. Besides describing the Participatory design process, this chapter details a redesign of the resulting artefact and a usability study meant to evaluate and refine said prototype. From this process, several key findings ensued, mostly regarding how game designers would interface with AGE appropriate it.

6.1 Participatory Design of the AGE Interface

At this stage, the main goal of the research was to clarify user needs and requirements for AGE, and design an interface with these in mind, adapting its functional working as needed. This section describes two participatory design sessions where eight different stakeholders designed a paper prototype for the user-interface of the tool. Participatory Design (PD) is an approach to design where end-users are placed in a leading role in the process (Sears and Jacko, 2009; Schuler and Namioka, 1993).

Given that the goal was to provide game designers with a tool that they would find useful for their creative tasks, providing them a key role in the design of this tool seemed an obvious fit, as their needs and wants should guide the design, with the added benefit of them being apt designers in the first

place. Hence the choice for Participatory Design. This design process served a dual purpose: a) designing an interface proposal from the perspective of its prospective users and b) providing a context to study how design practitioners would conceptualize, understand and interact with this novel approach to game design. In effect, this allowed us to start answering research question 2.

6.1.1 Methodology

Out of all strands in PD techniques (Crabtree, 1998) we chose Prototype Sessions, because it allowed the end-users to define, by themselves, how the end-application should function and interact. Prototype sessions provide a grounded environment where end-users can physically manipulate, create, edit, simulate interactions, and discuss low-res prototypes of the intended artefact.

The advantages in employing a Participatory Design approach are two-fold:

1. to quickly adjust this as a tool that can benefit their design process,
2. to start a preliminary study on the nature and impact that using such a tool could have in a game design context, by studying how designers discuss the prototype.

Because our research also concerns the study of how design processes can be affected by PCG tools, it is only natural that a design technique be used as a means to obtain data, given that one of the forefront subjects of design research concerns design praxeology (Cross, 1999). Furthermore, design itself can and has been used as a way of providing data for research activities (Zimmerman, Forlizzi, and Evenson, 2007; Zimmerman, Stolterman, and Forlizzi, 2010). Furthermore, getting game designers to design an interface for PCG-enabled approach to game creation seems an adequate and desirable context to study the way in which this new technology can be explored. It warrants that inspecting a design of this type will make explicit their mindset for this procedural approach, elucidate designers' intended uses for this tool, as well as the potential challenges and virtues in using it in their design practice. Thus, carrying out a Participatory Design process and then inspecting it can provide data and insights that lead to advance our understanding of interacting with the AGE approach.

Besides the participants designing the interface by means of paper prototyping, audio recordings from the exercises were used to support revisions and

discourse analysis, allowing further insight into users' perceptions. To inspect these Participatory Design sessions, we used Content Analysis applied to participants' discourse during the design exercise, by means of open coding by a subjective coder (Lazar, Feng, and Hochheiser, 2010). An initial pass of all audio recordings was done in search of key concepts that served as conversational themes, after which a new pass was done for the coding proper. A third pass was then done to validate and correct any coding errors. Naturally, forms of qualitative analysis are susceptible to biased interpretations and subjective manipulations (Lazar, Feng, and Hochheiser, 2010); more so in this case, given that only one person – the author – performed the analysis. The finality of any conclusions herein drawn must be tempered by this fact, and no claim of generalization can be made. As the purpose of this analysis was to complement users own prototype – by way of providing critical insight into the rationale behind their expressed needs and specifications – guarantees of objectivity were considered less important.

The aim here was to guide the design grounded on user derived data. Both quantitative and qualitative perspectives were used in Content Analysis; quantitative metrics (referring to the number of occurrences or time taken discussing a given topic) are used to identify and highlight major themes in discourse, while textual records are analysed in detail to provide minute semantic detail. Furthermore, Relational Analysis was done by searching for occurrences of disparate concepts in time windows of one minute. So, whenever two concepts overlapped, they were counted once (whether or not they repeated); from this quantitative measure, we then analysed the relational content of the most relevant pairings so as to interpret how subjects were associating these concepts, and view this in light of the design exercises. Conclusions were drawn from the results of this analysis.

6.1.2 First Participatory Design Session

The purpose of the first PD session was to design a first interface for AGE based on its algorithmic approach. One designer and a computer engineer were designing and developing a simple platform game, and their prototype was at a stage where AGE could be used. The designer had a background in multimedia and videogame design, with a few published projects; the engineer was a recently graduated programmer with an interest in game development, but whose contact with the discipline was limited to college courses. The two

creators had a basic prototype, but had yet to achieve the intended player experience. A 15 minute explanation of what this PCG approach was meant to achieve was given to these designers, and then they were tasked with creating an interface for it. It was suggested they design the AGE interface in a way that could help them with their existing prototype.

No mention of how AGE worked in algorithmic terms was given. Designers were only asked to design an interface in a way that allowed them to define the design problem (including definition of game features and target indicators), for the application would then 'magically' resolve it. They were told to draw the interface on paper, making use of office supplies (a similar set-up to (Muller, 1991)) whilst describing its workings orally, whenever possible making examples out of their own project. We expected that by grounding the exercise on their design process, they would provide the best solutions for their current work context (as per Crabtree (1998)).

Coding was divided into 3 major categories: a) mentioning of data sources, b) references to paper UI elements and c) references to participant's game design. In terms of data, several experience evaluation methods were mentioned, and so we subdivided counts in respect to the specific data type which they referred to. So, whenever a participant mentioned a game-play metric (e.g. *"game-play time"*), we counted one reference, the same for biometrics and subjectively determined qualities.

Because the platform is focused on varying game features, every mention of these was counted (e.g. *"power-ups spawn rhythm"*), under the category **features**. **UI** related terms were divided into sub-categories: a) **platform configuration**, or how data is inserted/edited/visualized in the application (e.g. *"a button, a thing to add a feature"*), b) **application structure**, how the program should be structured in terms of components and screen flow, (*"another tab where we could map all features"*), c) **data visualization** for how experience data should be processed visually when shown to users (what data transformations, statistics and graphic plots should be shown and interacted with) (e.g. *"you want to see a boxplot or a histogram"*), and d) **UX**, references to usability and other user experience considerations (*"I don't like the interface (...) in terms of flux and movement"*).

Game Design is the area we were most concerned with analysing. We extracted 3 main categories in discourse, relating to the approach: a) **optimization** of given experience goals, i.e. searching for the right set of features (e.g. *"what is*

the optimal number of obstacles"), b) **experimentation** of different game variations (e.g. *"test all possible combinations of possible obstacles"*), c) direct reference to **player experience** qualities (e.g.*"power-ups influence the level of challenge"*). Later in the process, we also registered what we call problematic events (for issues of comparison with session 2, to be further explored in the next section). A type of **terminological misconception**, registered whenever a participant used the wrong word for a given concept – e.g., describing features as if they were indicators. Also, the number of researcher directed **queries** that were made (implying difficulty in grasping certain concepts), and expressions of doubt on either the working of the approach – **doubts (approach)** – or its intended goals in terms of the creative design process – **doubts (goals)**.

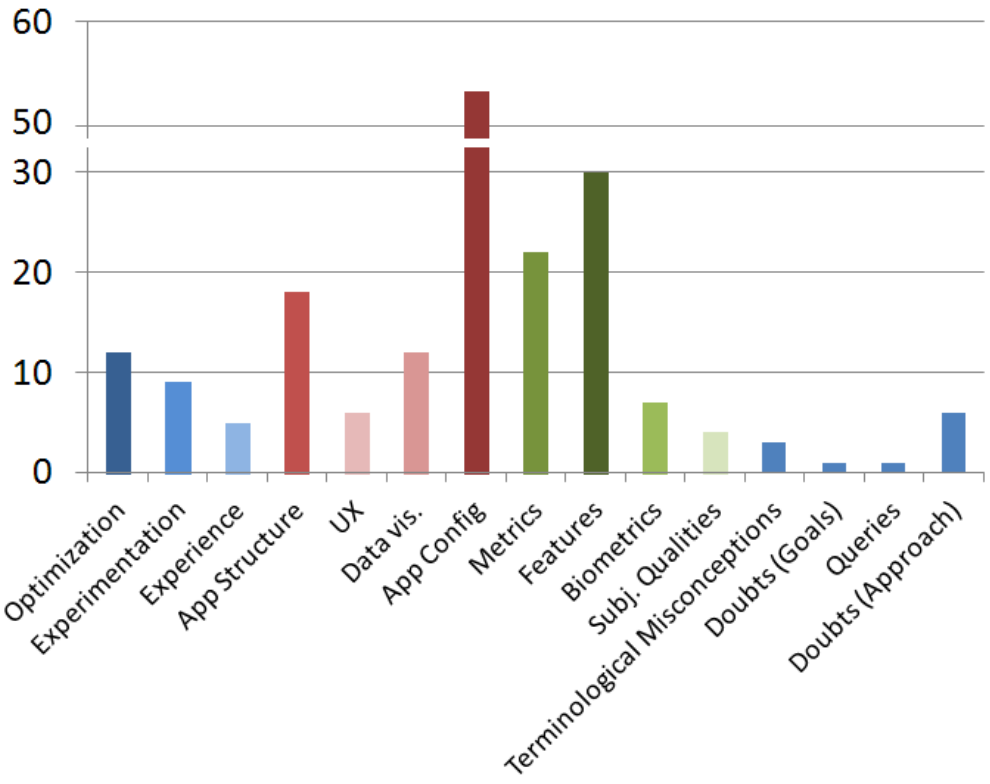

FIGURE 6.1: Counting of references. Blue refers to Design topics, red to User Interface topics, green to Data topics and violet for problematic events.

First Session Results

Figure 6.1 shows the event count for each topic and figure 6.2 shows a visual representation of a Proximity Analysis metric: the number of co-occurrence of concepts in one minute windows. The session was dominated by a focus on screens for inserting/editing data into the PCG tools, as the high number of **App Config** concept counts attest to. Further evidence is that the most prevalent

concept pair occurrence was that of **App Structure** and **App Config** with 16 counts. Example dialogue: *"navigating through the program to see what we wanted to alter, add, features..."*. This is further emphasized by high counts of two data concepts (**Metrics** and **Features**) and two very frequent co-occurrence pairs, **App Config** plus **Metrics** (11 counts) and **App Config** and **Features** (9 counts) where subjects focused discourse on how to get data into the application, e.g. *"a button I could use to increase or decrease the rhythm of power-up generation"*. We find a significant number of co-occurrences with pairs **App Config** with **Terminological Misconceptions** (6) and **Features** with **Terminological Misconceptions** (7); as while describing the process of inserting/editing a specific type of data subjects mistook one type of data with another (e.g. *"[subj.A] What you're calling metrics are features. [subj.B] Oh, right, yes."*).

Though the main focus of the approach is on finding the artefact that fulfils a certain experience (or **optimization**), the case for experimental testing of different cases was put forth, with no mention of target values. The number of events mentioning this approach was significant (almost the same as those of **optimization**), and when queried if this experimentation should be a use case to be realized before optimization, one participant replied that *"yes, something along that line"*, an *"exploration"*, matching the other participants previous comment on the alternative of having a use case where *"[experience] indicators are the output, and features the input"*. This means that this approach needed to be supplemented by a no target experimentation phase, where designers simply test out different variants of the same game.

Also, participants struggled with finding appropriate values for target indicators. The pair **App Config** and **Optimization** (13 occurrences), shows that there was a sequential order in the mindset of the process, starting with editing the data and following with the optimization of these values, *"For all these features, we can establish [here] their boundaries and values, and in that case, we then either optimize the rhythm or..."*. But optimization was not, as would be expected beforehand, a considerable focus of the session, and was only mentioned 12 times, and only in abstract, with no target values associated. This suggests that designers struggled to find meaningful values for certain indicators *a priori* without data.

Furthermore, the focus on data visualization, even though not dominant, speaks of the importance of supporting visual data analysis, irrespective of whether or not optimization goals are met. This also reinforces the need for use of the platform for exploration purposes. An underlying concern with

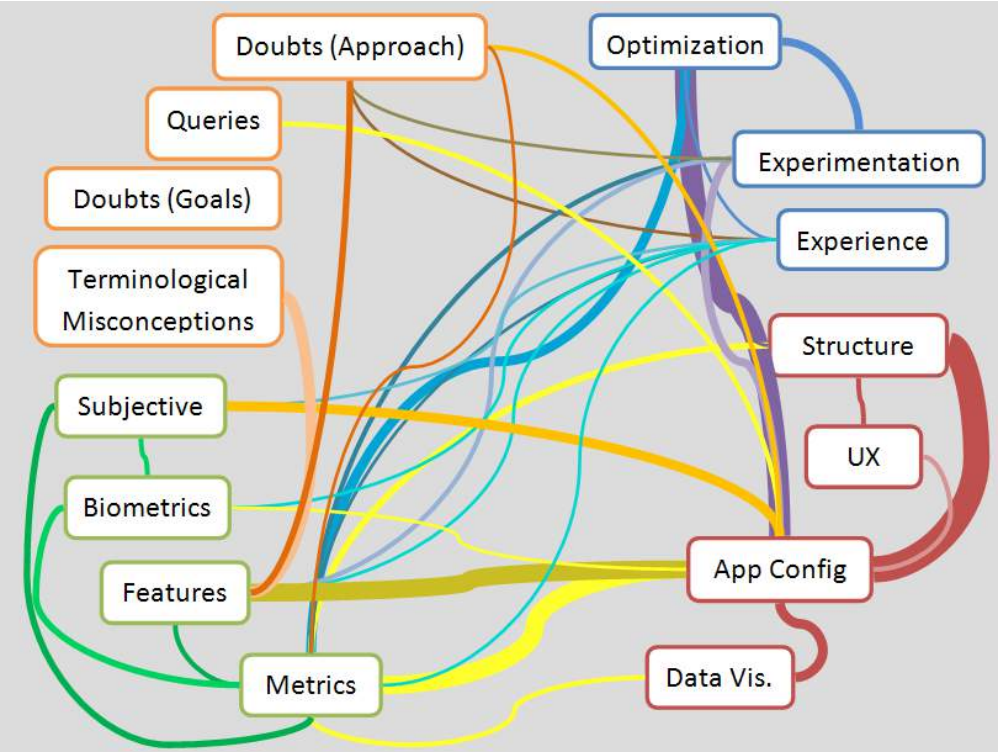

FIGURE 6.2: Figure shows bubbles for all the concepts that were codified during content analysis, with coloured links signalling co-occurrences in one minute windows. Thicker lines mean more co-occurrences.

procedural tools is that they take away control from authors, and here, designers stressed the need for access to data that can inform of what exactly is happening to players, both before committing to an agenda (presumably in reference to the exploratory use case), and after optimization.

Other aspects of the relationship between categories include the association of all types of data (**Metrics**, **Subjective**, **Biometrics** and **Features**) with the **Experience** category (2 counts for each link), signalling that subjects tried to bridge these concepts. And also of note, the preponderance of 4 different concepts in linking different subjects: **Optimization**, **App Config**, **Features** and **Metrics**. We had expected that a grounding on a real game design case would channel actual design issues into the design of the application's UI. Yet it seems very few references were uttered referencing the game in question (2 counts) and its player experience (5 counts), though all mentions of variables and metrics were taken from the context of its design.

6.1.3 Second Participatory Design Session

The second session was of a different nature. Because there was already a UI proposal, the second session was meant to iterate on that first prototype. We realized a full PD workshop with 8 members of a game research laboratory, including the two members of the first session. All participants had a vested interest in the development of a platform for testing game prototypes. Some would like to use the platform for commercial and creative ventures, as a way of designing and evaluating their own projects; others would use the platform for research purposes in the lab, as a way of designing and evaluating games for future study. 7 participants had experience in game design and development, and a background in either Computer Science or Design, and the 8th member researches Learning and Games.

Apart from the members of the first session, the team included one programmer who lectured and researched game design, having experience leading several game development projects, with commercial, research agendas or both. Two members with a programming background had designed one game before, and were mostly interested in sound design and music composition for video games. There was one programmer who worked as a game designer and developer in his own company, and had experience in creating several games; at the time, he had one published project. Another was a computer engineering masters student that had designed a couple of games for courses. Finally, the eighth member was a post-doc with a Education Science background, that was at the moment, researching left-field approaches for gamification.

Admittedly, the sample population was far from the ideal – which would be composed solely of experienced game designers. Still, we consider it marginally representative of the target population for this application. The session was started with a 27 minutes introduction to the platform, its current use cases and UI prototype. On account of the high number of participants, these were then divided into two groups, each discussing and iterating the interfaces separately for about two and a half hours. Afterwards, the two groups got together and shared their observations, discussing results for two and a half hours.

In the second session, we counted both number of events and measured how much time was spent on certain discussion themes. The reason for this is that, because of the large number of participants, discourse quickly lead to long winding discussions surrounding a single topic, and counting these as singular events would result in a small number of events that could not

represent the importance they took in the session. In terms of events we added coding for expressions implying difficulty in understanding some aspect of the platform, **Trouble Understanding**, and **Functional misconceptions** regarding the platform working.

Discussion themes include **UI**, which refers to all discussion of the interface design. **Composed Features**, a topic that revolved around the possibility of establishing complex game parametrizations, where each feature could vary in relation to a distinct one, *"doesn't it make sense to have a feature type that is relational? [A feature] in function of another feature"*. **Preset-Features**, the proposal to add of a bundled set of default artefact features.

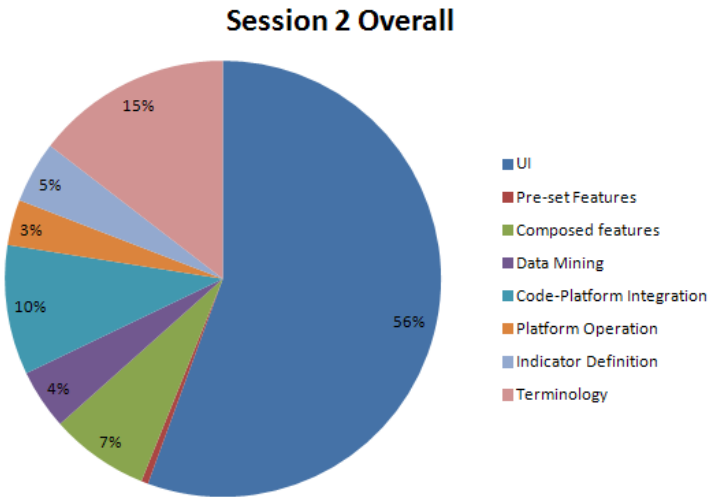

FIGURE 6.3: Overall division of net discussion time according to theme.

Discussions that concerned how the game-artefact and the procedural platform should be integrated were coded with **Code-Platform Integration** (example:*"when you add [a game], it generates an id and a hashcode you place in your code, to export data [to the system]"*). Also, **Data Mining**, referring to a proposal for the use data mining techniques for harvesting of meaningful indicator data, *"imagine a variable [from the metrics logs] that the [Artificial] intelligence detects – here is something going on in this parameter –, so we should take this to an observational plane"*.

Debate arising from doubts on how the platform should operate were tagged with **Platform operation**, for example, this: *"what allows you to say that this particular set of features is ideal for this game?"*, sparked a debate, because that participant still had not realized the 'ideal' could be defined by designers. Another discussion was referring to when to define Indicators, if before or after running a

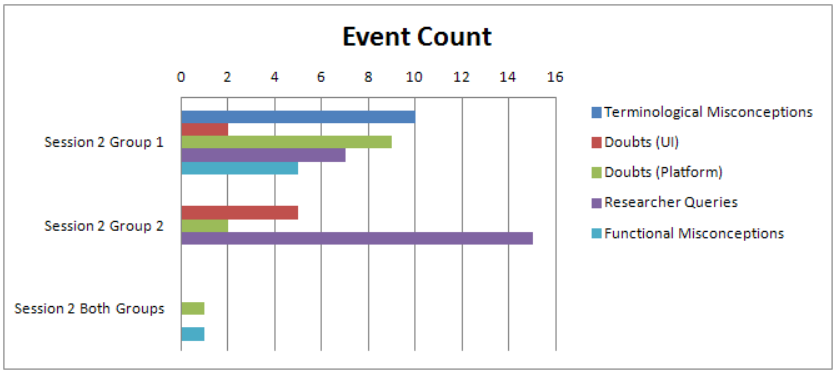

FIGURE 6.4: Counting of events relating to difficulties in understanding the procedural tool. Divided into Session 2's three parts: discussions inside Group 1 and 2, and discussion between both groups.

prototype test, **Indicator Definition**, example: *"when I define indicators a priori, it limits my possibilities of exploring results"*. And also, strings of discussion focused on attempting to pinpoint **Terminology** or come up with conceptual definitions of concepts, e.g. *"a game can have several experiments, each experiment has several tests..."*.

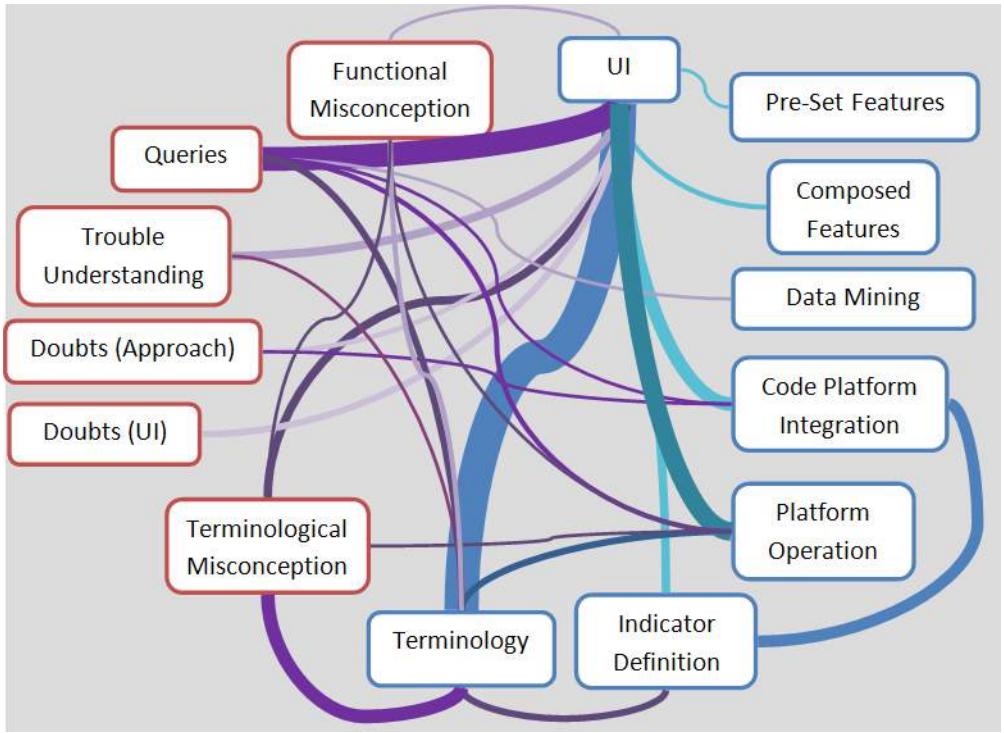

FIGURE 6.5: Co-occurrence count for the second session; same format as 6.2.

Second Session Results

As expected given the goal of designing the interface, 55.6% of discussion time concerned the **UI** topic and it was also the main focal point of links in figure 6.5. More surprising is a non-negligible (9%) portion of time spent discussing how the platform operates, which is symptomatic of the approach's concepts and functioning being hard to comprehend. This is confirmed by the high number of problematic events: 10 **Terminological Misconceptions**, 6 **Functional Misconceptions**, 7 **UI doubts**, 13 **Platform Doubts** and 22 **Researcher Queries**. The almost 15% of time spent on discussing terminology and underlying concepts seems of considerable importance in this respect as they imply a great deal of conceptual confusion. Highlighting this effect, terminology discussions lead to Terminological Misconceptions 8 times.

In terms of which concepts subjects struggled more in pinning down, they were either the types of data (features), i.e., *"aren't these indicators [referring to features]?"*, or its procedural aspects (what should constitute a 'test', for instance). To improve this, we believe a concerted effort must be placed on making these concepts more rapidly understandable and intuitive, so as to avoid ambiguities. In the case of procedural concepts, this seems even more relevant, as when the experimenter asked subjects to create working definitions for these aspects, they could not do so. As one subject said: *"there is a conceptual base here that has yet to be defined, and it is very important"*, and *"the language is not completely defined"*. A solution here would be to define a solid ontology, with accurate definitions and relations, so that there can be a more effective appropriation of the application. The lack of a powerful metaphor that could communicate this future design approach with existing practices was keenly felt.

Nearly 10% was spent on determining how the approach should interoperate with games' prototype code. While no consensus was found on the best solution; the preoccupation in this discussion was how to make the process easy and seamless. Four major design proposals were forwarded during the sessions. The idea of **Complex Features** (features that compounded several artefact features) was heartily discussed twice, sprouting a great deal of discussion (7%), but with most members either opposing the idea or abstaining from commenting it. Time spent on **Pre-set Features** and **Data Mining** (1% and 4% time respectively) seems negligible, and in fact, none of these ideas translated into actual specifications. In all these cases, participants betray a desire to have an easy to set up application, that has some way of configuring solutions all by itself. Naturally this is very

much relevant to the PCG tool design, and one of our key concerns.

Finally, the **Indicator** discussion (determining when indicators should be defined, whether before starting a test or afterwards) did not take up as much time as other topics (5%), it actually lead to a new specification. In the end of the conversation, there was an agreement that new indicators should be possible to define both before and after a test has been run, a compromise between the initial proposal that defended *a priori* definition (essential for optimization) and the new proposal, that was post-test definition. This reinforces the need for an undirected, explorative content generation approach (where indicators are undefined), just like the exploratory use-case from the first session. Thus, this became part of the approach's specification.

6.1.4 Participatory Session Results Discussion

AGE's goal is to give game creators a tool for more effectively crafting their game-play experience agenda. The two design exercises herein described, where we asked interested designers to design a UI for this approach, were meant to not only provide us with a working design for the UI – which was achieved – but also reveal their needs, wants and preoccupations in regards to this approach. Besides the actual outcome of the UI design, as well as certain key revisions to the prototype's functioning, we can sum up several crucial aspects that came out of these sessions.

Procedural Content Generation needs a metaphor.

Before this PCG approach can be operationalized in a game design scenario its functioning needs to be easier to understand by its users. There was a consistent struggle from subjects to understand the nature of this tool: from the arise of queries and doubts, to outright terminological misunderstandings, there is ample evidence of difficulty in grasping the platform's goals, concepts and resulting game design process.

This process highlights the proposal by Johnson and Henderson (2002) that before designing a system, one must design the conceptual model that underpins it and discuss and make it consensual in the design team. AGE had a conceptual model before these sessions; however, that model was not easily appropriated by its intended users, because it was too complex and its terminology distanced it from its domain. An excess of concepts – features, indicators, tests, etc. – without

an intuitive semantic framework (that could not be mapped easily to designers' domain), as well as the approach's complex mode of operation, presented a high barrier of access to users.

Up until this point, we had not clearly considered that AGE's working mode and components needed a communication strategy – a carefully chosen set of names and metaphors that could help users understand how to use it. This lack of forethought proved costly, as the lack of terminology severely hampered users capacity to quickly understand both AGE and its interface design.

A new approach to design requires a new design language. Our approach to solve this is to propose a simple metaphor to encompass this approach, and then make sure both the application's terms and logic are coherent with it, so that users should have less difficulties to understand how AGE works.

Exploration before Optimization.

The other major difficulty that participants had was in how to define the game design problem and solution according to the metaphor of this procedural approach. It requires a reversal of the traditional game design flux – to decide on experiential qualities before game's material features – and it was never fully incorporated in subjects speech. In the first design session no values for experience indicators were ever discussed and in the second, little to no references were made on actual design cases and agendas that could be fulfilled with this approach. This puts forth the question of whether or not it is practical for designers to reverse their mental processes in this way.

On both sessions, one participant made the case for a new use-case, either to exist separate or to be carried out before an optimization, where the tool would be used just for exploration of the design space, in search of data on the landscape, mapping out the relationship between different features and player experience.

The UI proposed by participants contains one window solely dedicated to presenting results from game-play sessions, using plots and tables, and part of the discussion, in both session 1 and 2 was in reference to this topic. Thus, an integral part of this application needs to be focused on how to explore game-play experience data that can inform the design process. In this way, designers can have exploratory phases to map out the design space, before committing to a design agenda that they want to optimize.

6.2 Interface Prototyping

Based on the prototype that came out of the PD sessions, and taking into account the results from the discourse analysis, a new iteration of the design was made.

6.2.1 A New AGE Design

The first revision to the AGE approach was the addition of a new use-case, that of exploration, as during the design sessions, it was forwarded by participants as an interesting use of the application. Besides this change in use, there were two core structural differences in the design: the first is the introduction of a new metaphor for the application and the second is a consequent change of linguistic terms.

Originally, the terms features, metrics and indicators, quality and candidates, experiment and test were used, and many of these led to interpretation problems or simply lacked a common working definition; on hindsight, the terms do seem extremely vague and open to interpretation. We used this design revision as an opportunity to find terms that could be more intuitive for game designers, as well as more harmonious with the new metaphor. Finally, this new design also includes minor additions to the interface aiming at ensuring core functionality (e.g. having a login screen, a button to control the algorithm's deployment, etc.).

The new metaphor is that of a game test-bed. In this new framework, designers define how candidate solutions are generated and evaluated, by establishing **Game Variations** (formerly referred as 'features'), and **Design Goals** (which replace 'target indicator values'). The procedural content generation algorithm creates solutions based on an initial Base-game; this base-game's variables are manipulated (or varied) so as to find new game configurations; these are named **Game Permutations** (as they are achieved through a (conceptual) permutation of values in the original game). **Game Variations** are the rules that determine the boundaries and mode of variation for each of the game's variables, meaning each **Permutation** is defined by the set of values its game variables hold.

A Design Goal test defines one test rule for how to evaluate each Permutation. It has one condition and an associated score, so it can be translated to a simple phrase such as 'if the number of player kills is greater than 15 (this is the test condition) then the candidate gets 100 points (this is the test score)'. The PCG

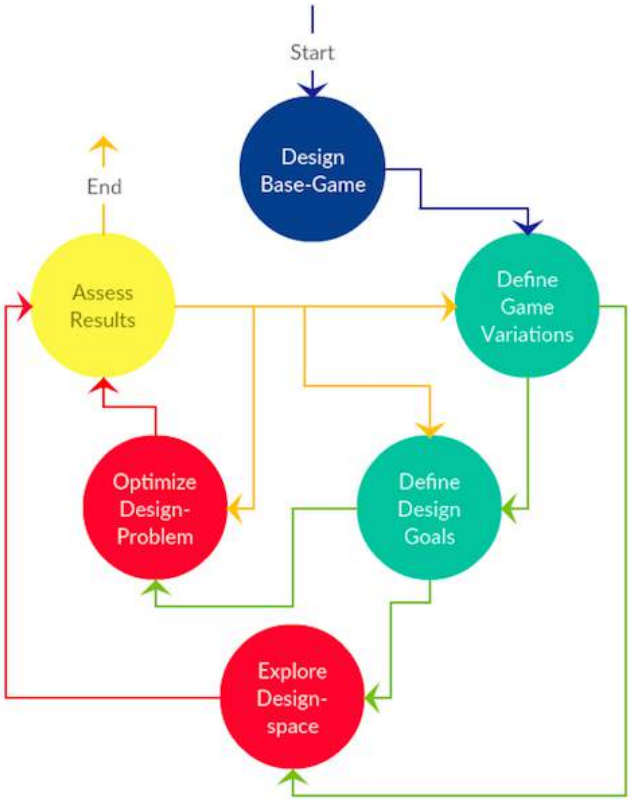

FIGURE 6.6: AGE game design process, task by task, as proposed during Participatory Design Sessions and using the updated terminology.

algorithm then generates, tests and eventually finds solutions that have the best average score for all Design Goal tests. Figure 6.7 exemplifies AGE's working for one candidate, using the new terminology.

We also gave greater granularity to the application's lexicon. In the original version, the term feature was used interchangeably for both game feature and the rule for varying a game feature. Thus, we changed the term feature to the less ambiguous Game Variable, and also employed Game Variation to denominate the variation rules for Game Variables. The range between which each variable varies is called the Test Range. Similarly, Game Metrics and Indicators were also used interchangeably, and though very specific, they require in-depth knowledge of the area to accurately pin down. Thus, we named these **Target Experience Qualities**, which while more extensive, should make it more easily understandable for non-experts. As to the actual optimization of these qualities, we proposed a new collection of terms. The range for a specific Experience Quality is now named the test range, and when a quality is in that range it gets attributed a (Test) **Score**, attributed according to a **Score Function**. And each

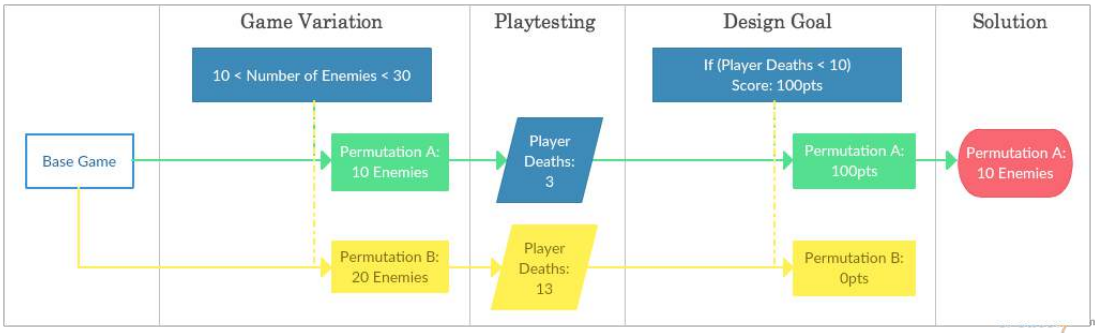

FIGURE 6.7: Visual representation of a single AGE optimization cycle, using the new terminology, with 1 Game Variation and 1 Design Goal. The base game is taken, and according to Game Variations, two candidate solutions are generated, A and B, with 10 and 20 enemies respectively. These are then play-tested by players, who die, on average, 3 and 12 times, respectively. According to the Design Goal, candidate A gets 100 points, and B no points whatsoever. The best solution is then attributed A, the candidate with 10 enemies.

objective in the optimization is named a Design Goal, as this makes it clearer that the objective is to achieve a certain target for each experience quality.

Other terminological issues had to do with the nature of each game project, and what constituted an optimization test. We hence created a new concept by using the term **Design Problem** to encompass a set of Design Goals and Game Variations, as together these define, for one game, a particular design problem to be solved (and consequently, the search for one design solution). As to the procedural entities, we proposed **Experiment** as each optimization project, itself comprising of a combination of one prototype with one design problem. To run each experiment, designers conduct **Tests** in which a full design cycle is completed: the system evolves Game Permutations, players play the game, and results are assessed. One experiment can have several tests, and whilst at any given moment there is but one Design problem, it can be altered through time, as the designer fine tunes what his design is aiming for.

Our expectations were that the use of this metaphor would make several aspects of the AGE solution finding process clearer, and therefore, make AGE easier to use by designers. This new metaphor makes it clear that AGE works like an automatic generator of solutions which are tested until a good solution is found. Like in a test, there has to be some sort of right answer for the design, a model of what the game should play like. And by providing semantically richer nomenclature, we also hoped to avoid misinterpretations; features and

indicators were more simpler, more abstract terms, while Game Variations and Design Goals signal more loudly their function in the AGE approach.

The change in metaphor, terminology and interface revision, became part of a new tentative design that we aimed at testing. Because the PD sessions afforded a prototype that did not solve the core comprehension issues in our approach, it seemed natural that this design amendment had to be proposed for potential users to evaluate it and adopt or reject it. The first step with that goal in mind, was to make a thorough incorporation of these new concepts into the existing design prototype. This was not a complete redesign, merely a new iteration introduced by us. Whenever possible, the original was maintained as faithfully as possible, and changes were always made while trying to keep the integrity. To validate it, we needed to test this new prototype. We chose to do so on two levels: both in terms of its usability, as well on the intuitiveness of the new terminology.

6.2.2 Usability Evaluation of the AGE Interface

We implemented a first digital version of this prototype and then elaborated a task walk-through of all its main functionalities. We presented the AGE approach and its core concepts in a very short, open to questioning 5 minute maximum presentation, and then asked 8 subjects to complete this walk-through. This was divided into 6 different tasks:

1. Register and Login.
2. Create a new Game Variation.
3. Create a new Design Goal.
4. Edit a previously created Design Goal.
5. Setup a Test.
6. Check current Test Results.

For each task, we measured whether or not subjects successfully completed each task and the time it took (if successful). Also, once each subject finished a task, we queried on their perceived difficulty for the task – "I found this task easy to accomplish" – on a 1-5 Likert scale, and then assessed the comprehension of the new terms that were involved in the task. This was accomplished, firstly, with a simple question to assess subjects' confidence on their understanding of the terms, taking into consideration both the small presentation, the application, and its help function – "I understood what the term X refers to." – again on a 1–5 Likert scale.

Then, to measure the accuracy of subjects' understanding we verbally queried them on "What is X?", allowing subjects to discourse on the nature of the concept. Because of the concept's complex nature, this line of questioning, although planned, had to be dynamically adapted so as to accurately track subjects understanding of the concept. Follow-up questions meant to disambiguate subjects discourse were on the lines of "what is the difference between a Game Variable and a Game Variation?", and "how many unique vales can be derived from this [pointing to on-screen example] Game Variation?".

Two researchers, including the author, carried out these proceedings, and only when both agreed that subjects had correctly replied to the entire line of questioning would a response count as correct. If the reply was not deemed entirely correct by at least one of the researchers, it would count as incorrect. The reduced number of participants in this study does not allow generalization of its findings. However, studies show that for usability purposes, small numbers of users can still, on average, detect a high percentage of errors (Faulkner, 2003), which is exactly the aim of this exercise.

Usability Test Results

First, in respect to terminology issues. We tested the six main new terms that came with this prototype's design. No term was understood by all participants, at least one of them always misinterpreted one term. That said, it seems reasonable, considering the degree of complexity in the approach for this to occur, especially given subjects had very little time to accustom themselves with it and its terms. As such, we established a maximum number of one erroneous answer per term as acceptable. This leaves two terms (Session and Game Variable) with two subjects misconceiving their meaning and one term with three (Game Permutation).

In terms of Game Variable, we think the problem laid in the similarity with 'Game Variation' rules; when queried on what a Game Variable is, one subject said *"Haven't you asked me that already?"*, clearly referring to the previous question on the nature of a 'Game Variation'. Naturally, these are intimately related terms, some overlap was to be expected. As to 'Game Permutation', one subject relayed that the term suggested an exchange of values – *"Permutation means exchange (...) The definition of permutation is exchange"*. The term permutation then seems needlessly convoluted, and perhaps Game Candidate would be the better choice. In the end, all these three terms need further revision, to make them more communicable.

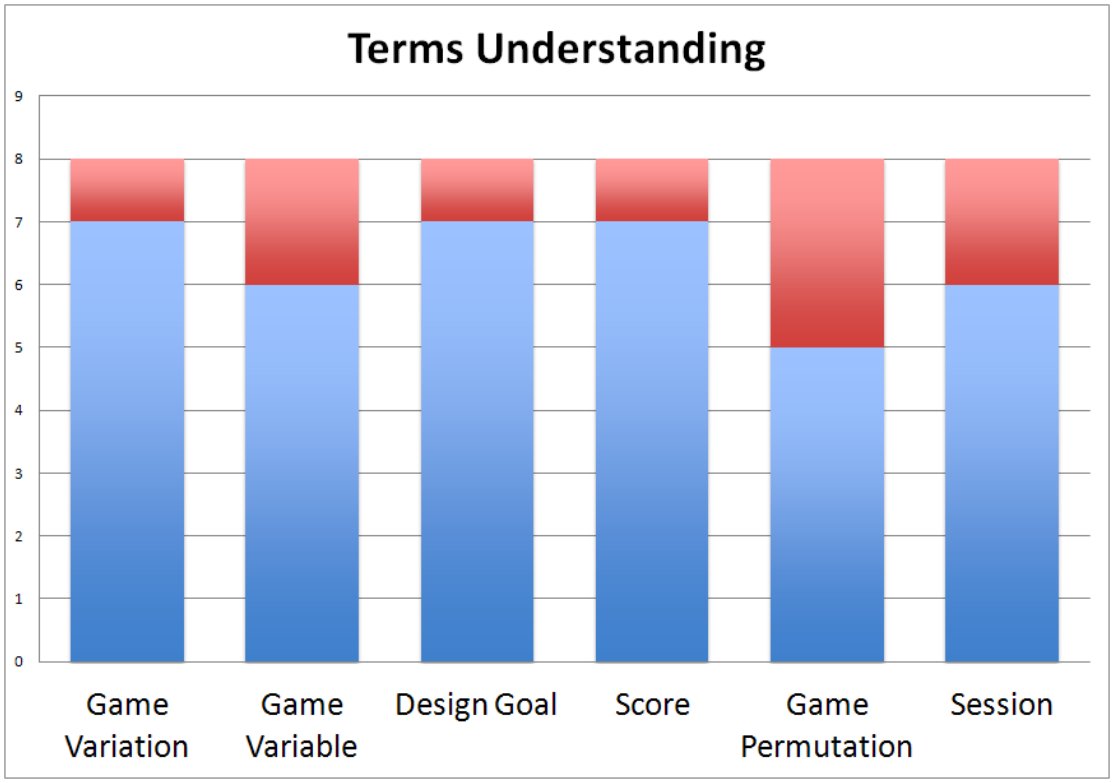

FIGURE 6.8: Results from the usability test in respect to subjects understanding of each of AGE's linguistic terms; in red are the total number of subjects which misunderstood a given term. At least one subject misunderstood each of the terms, and Game Variable, Game Permutation and Session went beyond that already conservative standard.

Curiously, subjects' confidence in terms was high, even for misunderstood terms. The average was always above 3, and there were only three instances of subjects giving a confidence level below neutral, two subjects for Design Goal, and one for Score, coincidentally, referring to terms that passed our maximum error threshold when it comes to correct replies on its meaning. This suggests that these set of terms do not, at least, induce major incomprehension in part of subjects. They are meaningful and inspire confidence, even when subjects misappropriate them. Although not an ideal scenario, it seems to indicate that the test-bed metaphor provides an adequate contextual mental framework for users.

All things considered, we propose revisions to the terms Game Parameter (as opposed to Game Variable), Game Candidate (as opposed to Game Permutation) and Play Session (as opposed to Session), for future design iterations. This should help solve the remaining terminological misconceptions.

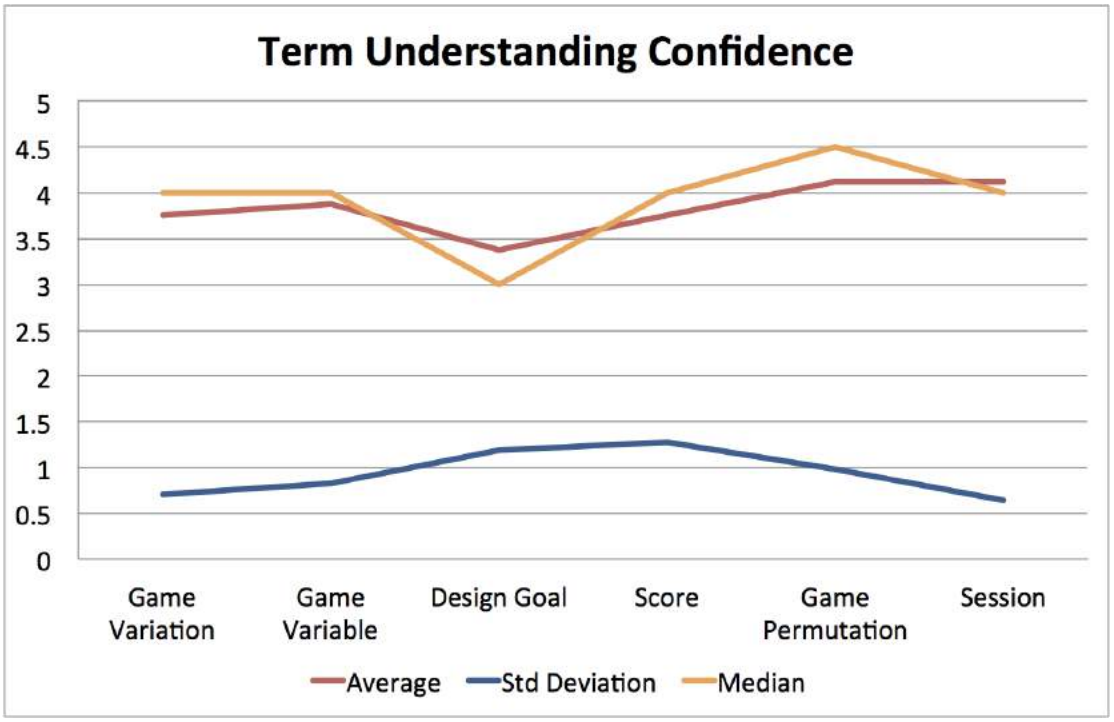

FIGURE 6.9: Average, median and standard deviation of subjects' reported understanding confidence for each term (on a 1–5 scale).

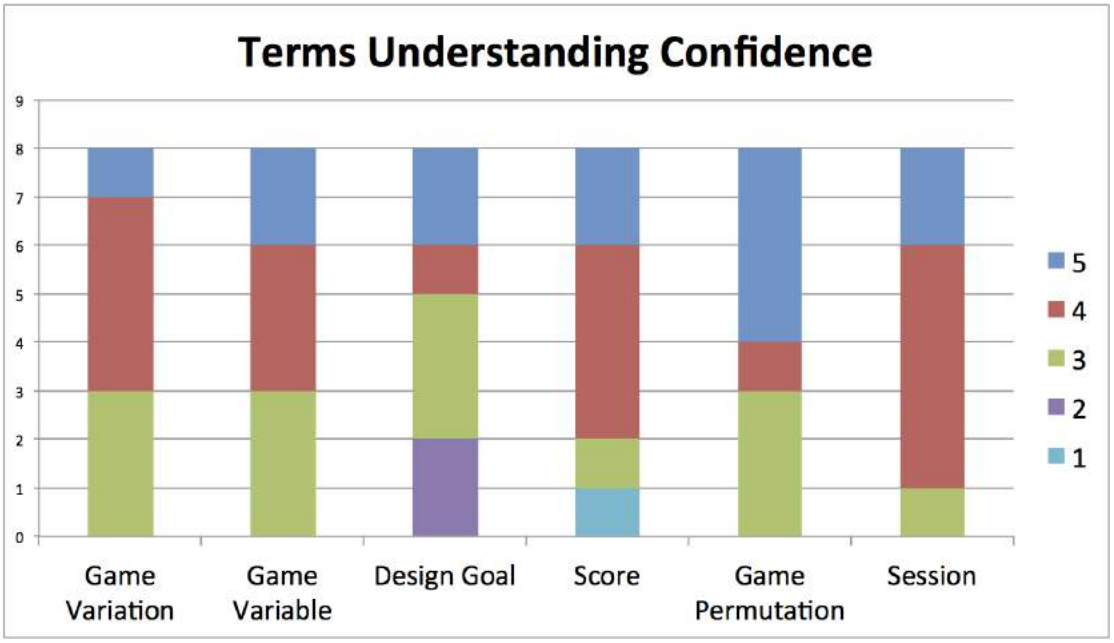

FIGURE 6.10: Each series corresponds to a given confidence level, showing how many subjects reported it for a given term. Only Design Goal and Score have one or more replies below 3. Curiously, there seems to be no obvious overall relationship between term understanding and confidence.

Design Proposal	Design Revision 1	Design Revision 2
Game Feature	Game Variable	Game Parameter
-	Game Variation	Game Variation
Target Experience Indicator	Design Goal	Design Goal
Solution	Game Permutation	Game Candidate
Quality	Score	Score
Session	Session	Play Session

FIGURE 6.11: How major concepts terms evolved in AGE's design throughout time. Note that some changes are more than merely linguistic, and actually involved changes on a functional level (resulting from the introduction of the test metaphor).

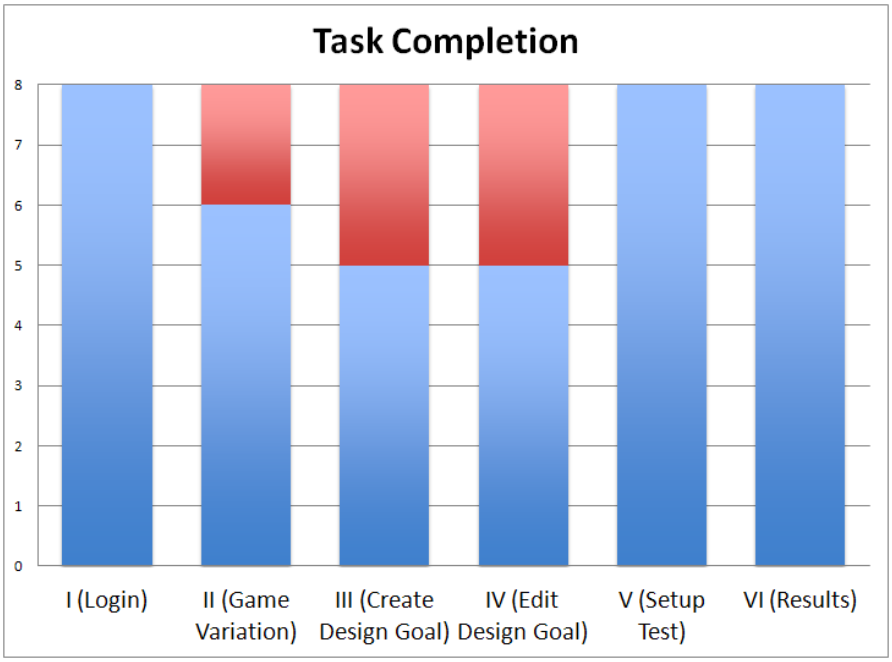

FIGURE 6.12: Usability test task completion. 3 Tasks showed issues with some participants, all related with Design Problem definition.

Task completion rate shows that the three more important UI tasks – Game Variation creation, Design Goal creation and editing usability issues. The severity of these seems to vary. Task failures referring to Game Variation creation resulted only from a misplaced UI button (the button to activate each game variation, was only present in the Design Problem menu and not in the Game Variation edit sub-menu); this has naturally been fixed to make the user interaction flow more intuitive and less prone to this particular problem.

As to Creating and Editing Design Goals, issues came both from a similar problem (instead of activating a Variation, users had to chose to activate an 'Optimize' button for Design Goals), and from one particular sub-task: indicator

formula creation and editing. This was the major hurdle in Design Goal tasks, and it becomes clear how big it was when looking at completion times. Out of 7 different steps in the Design Goal creation task, subjects averaged 45% of their time only in trying to complete the formula (standard deviation of 10%) in the editing task, comprised of three steps, the average was of 60% of task time for the formula only (19% standard deviation). And this, taking into account that five out of a total of sixteen formula activities, subjects were not even able to find the right formula and/or merely gave up before succeeding, therefore reducing some of the measured completion time.

The formula sub-screen is highly complex and was not sufficiently detailed during the Participatory Design sessions. Determining how to calculate an Experience Indicator from raw logged data is a task as complex as writing a formula in Excel or a SQL script. While the screen does not afford a comparable degree of power in its editing capabilities, it is extremely open to configuration. The task walk-through was purposefully created to test whether or not designers could use it – subjects had only a literal description of a formula's functioning (*"Create an indicator formula that calculates the time needed to complete a level (this is the maximum value of 'engine time')"*), and it was up to subjects to translate that into a formula and then insert it via the screen's form.

Despite the simplicity of the required formula, subjects struggled with the task. Our own view is that this would be a task normally relegated to programmers and not necessarily to be carried out by designers themselves. Still, the difficulty found by subjects was greater than expected. These two tasks referring to Design Goal creation and editing are part of the core tasks of creating a Design Problem; therefore, difficulties arising in these two need to be eliminated. Besides redesigning the formula screen, a concerted effort to find solutions for making it easier to create and edit indicators should be looked for.

Taken as a whole, our interpretation is that some aspects of the participatory designed interface are just too complex on account of the complexity of the AGE approach. While a formula screen of some kind is a mandatory functional requirement for AGE to be implemented, the way it is contextualized and presented to users clearly needed rethinking. The remaining issues seem easier to solve, where minor fixes to language and interface are likely enough to make the interface usable.

6.3 Discussion

The user interface created by designers, and then further refined by us with the test-driven metaphor, essentially answers how human designers can translate design problems into a computational format that can then be solved by procedural algorithms – the answer to the research question *How can game designers interface with procedural content generation algorithms so that they can solve game design problems?*. The proposed terminology is a work in progress, but already seems reasonably functional, and given new iterations is sure to solve the issues from the Participatory Design Sessions. The same can be said of the interface – a majority of users were already capable of using it, and given the corrections suggested by the usability study, it is sure to become usable by the target audience. Issues that remain result from what looks like a complicated interface, that is, in all likelihood, too convoluted for designer use; future design revisions must focus on streamlining and greater accessibility.

Participatory design data also began to shed light on the second research question – *"How does the introduction of AGE transform game design processes?"*. The suggestion of a use-case concerning Exploration means designers would use AGE to explore the design-space, and not necessarily just optimize a given game. The focus on results feedback also places emphasis on designers wanting to validate the process without letting the search algorithm doing all the work. Based on a literature review (Shneiderman et al., 2006), a set of 12 guidelines were proposed for creativity support tools (i.e. *"computational systems and environments that people can use to generate, modify, interact and play with, and/or share artifacts"*), and at the top of the list is *"support exploration"*,*"to emphasize easy exploration, rapid experimentation, and fortuitous combinations"*. Though the initial AGE design was not intended to support it, data from the sessions supports literature in that exploration should be a major thrust of the tool, irrespective of its original design. It thus becomes a major part of the AGE design.

With the end of this chapter we effectively closed the first complete design iteration of the AGE tool, which means it was time to study how game design would be impacted by use of the tool. Hence, the next chapter describes a case study design exercise with AGE.

Chapter 7

Case Study – Designing a Game with AGE

Given the complexity of the processes and information required to use this tool, and seeking to validate its usefulness as a tool for creative game design, AGE needed to be tested in a design context. To do this, we took the AGE prototype and prepared a case study to explore how designers could work with this approach.

7.1 Setup and Data Collection

To provide evidence of the how AGE could be used as a tool for support of the design activity, we wanted to find answers to the following research questions: Can designers define design problems using this approach? How can they define them and with what limitations? Are approaches to creative problems made possible through AGE ? How do designers appropriate the tool?

We started by selecting a base-game; we took an open-source version of the Dune2 game, and integrated it with the AGE tool's system. To expand the work done on section 5.3, we used a different prototype in a different genre. The logic of this choice was to experiment with new contexts, so as to avoid creating a tool that is biased to a specific game genre's design. By working on a different prototype, in a different genre, we are widening its potential usage scope and fine-tuning the design for it to be as problem-independent as possible.

We asked a game designer to use the Dune2 prototype and come up with a novel game design which he would desire to improve using AGE. This designer is a 31 year male, with an extensive portfolio of experimental games and interactive entertainment applications, and his background mixes informatics with

photography and design.

The set-up was simple: ask the designer to utilize AGE to evolve the Dune2 base-game; any game-design agenda would be valid, as long as it could be feasibly realized based on the base-game prototype. Were this a conventional game design scenario, the base-game and its integration with the AGE system would be implemented by the designers and developers themselves, yet this was not a feasible option for this experiment. For this reason, throughout the exercise the subject was asked to freely propose changes to the base prototype, detail any Game Parameters he wanted to be made available for exploration, and list data metrics to be extracted for analysis and/or optimization by the tool.

A total of 5 Experimental Sessions were recorded: first, a preparation phase without contact with the tool prototype. At this point, the designer inspected the existing prototype to acquaint himself with the base design. Then, he was given time to, on his own, come up with a concept for his game. Once he responded he was happy with his idea, he was queried to list any alterations he would like to either the base prototype or its integration with the tool. After this preparation phase, 4 sessions occurred where the designer could use the AGE tool to define his design problem and find a solution for it.

In between each of the four main sessions, a small group of players (varying from 1–6) would serve as play-testers, trying out game candidates evolved by the system. A total of 87 complete game-play sessions comprising of 662 minutes were logged. During each session, the designer inspected results, proposed design alterations and configured new rounds of play-testing with the AGE tool. During each session, an experimenter was present that interacted with the designer in 3 distinct ways: a) assisting the designer with the AGE tool interface (dispelling doubts, providing tutorials for key functionalities), b) detailing aspects of the base-game and confirming what design changes were possible, and c) querying the author on his decisions, asking for rationalization for his actions, and documenting the process in general. Because of the prototypical nature of the AGE tool, 'a)' was particularly important so as to smooth out the design process. Also because of this fact, the designer frequently voiced small additions to the results interface he felt he needed in order to analyse play-test data.

7.2 Methodology

Design sessions were inspected using a convergent mixed methods experimental design. We collected both quantitative and qualitative data: quantitative data was used to highlight major behavioural and attitudinal patterns during the designer's creative activity, and to identify critical issues with the use of the tool, and qualitative data to find his interpretation to issues he was encountering. Quantitative data includes user metrics that tracked all designer-interactions with the AGE interface, and a Creativity Support Index report (Cherry and Latulipe, 2014) filled by the designer at the end of process, so as to gauge his subjective evaluation of the tool. Qualitative-wise, we recorded all *in situ* interactions to allow us to track designers verbal expressions. Because we asked the designer to engage in a think aloud protocol, and queried him at several points in the sessions, we can extract discourse snippets that shed light on his interpretation of events.

A very modest form of content analysis was done on his speech, so as to quantify major design events, such as the number of design proposals he voiced during the sessions, or number of alterations he proposed to AGE. These were used in conjunction with quantitative data, so as to find major patterns. And finally, at the beginning and end of sessions, the designer was openly interviewed to provide further insights.

What follows is a presentation of quantitative results, meant to highlight key issues that occurred in the design sessions, and after it, is a detailed description of each session using the designer's own words. These are dissected in detail in the Discussion section.

7.3 Quantitative Results

Figure 7.1 shows major design events, registered both from the interface and after open coding of the designer's dialogue. From his speech we extracted the following categories:

1. **Base-Game Edits** are the number of communicated design proposals made by the designer so as to change the game prototype.
2. **Cut base-Game Edits** refer to editing the designer expressed verbally but eventually dropped from the existing design.

3. **New Game Parameters** – the number of times the designer asked for a Game Parameter to be added to the game prototype for posterior variation.
4. **New Metric** maps the number of times the designer asked for a new event during play-testing to be registered by AGE.
5. **New Tool Specification** highlights when the designer (though we were not expecting it) asked for changes in the workings of AGE.

Additionally, for matters of comparison we added some user-interaction events:

6. **New Game Variations**, refers to every new definition of a game variation to be used by AGE to create new candidates.
7. **Edited Game Variations** – how many times existing Game Variations were changed.
8. **New Design Goals** for how many design goals were created throughout the exercise.
9. **Edited Design Goals** for number of edits.

Note that designer never forfeited control over the game – avoiding optimization of player experience, and proposing a total of 6 direct changes to the base-game (as opposed to using Game Variations to evolve it). While he created four Game Variations and edited them 6 times, because he never activated the evolution, we assess that these were not deemed as that important. This tells us that the designer appropriated the tool mostly as a test-bed for exploration of the design space surrounding a hand-crafted prototype, and not, as also intended by its design, as a medium for semi-automatic improvement of the base-game.

Figure 7.2 shows how many events were triggered in the Design Problem page (where he established the design problem) and the Results Page (where he looked at play-testing data), per session. This provides an overview of how his work was divided in between design problem defining and play-test data viewing tasks.

We can deduce that there was a great effort in terms of defining the Design Problem in the first session (when there was no data to analyse) and to a lesser degree, in session 3. The designer then focused on player data analysis in all but the first session. Overall, it seems there was a predominance of the latter; with a total of 328 registered interface events in the Design Problem, and 415 in the Results page. A similar pattern emerges in terms of total time spent on these pages – 1h 46m 27s and 2h 19m 45s respectively. In both cases, data indicates greater effort was spent in analysing results as opposed to defining the design.

Event Count	S0-S1	S2	S3	S4	Total
Base-Game Edits	3	0	3	0	6
Cut Base-Game Edits	0	0	4	3	7
New Game Parameters	4	0	0	0	4
New Metric	2	2	2	1	7
New Tool Specification	1	3	0	0	4
New Design Goals	3	0	2	0	5
Design Goal Edits	7	2	1	0	10
New Game Variations	4	0	0	0	4
Game Variation Edits	5	0	0	1	6

FIGURE 7.1: Per-session counts of main design events. The top rows refer to data coded from the designers' speech, and the bottom rows to critical data extracted from his interaction with AGE (full description of these in section 7.3).

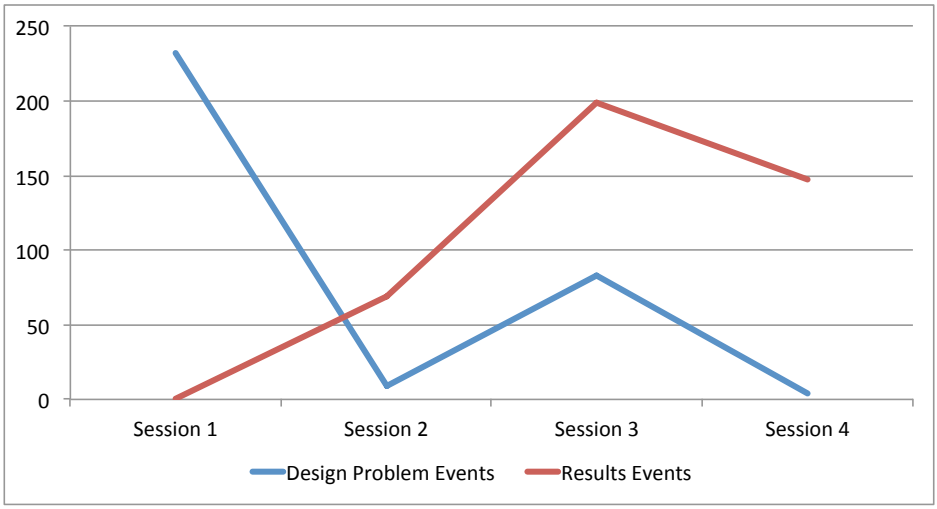

FIGURE 7.2: Aggregate of user interactions with the AGE tool interface.

Cross-referencing this data with the speech content, we interpret that AGE was used predominantly to test two design proposals (a first in session 1, and a second in session 3), and an active search for player experience data on these.

The third chart (figure 7.3) shows a more time-detailed picture within each session. All interface interactions were accounted for; however, because these total more than 100 different interface metrics, we needed to select a shorter set for visualization to be practical and meaningful. The criteria we decided upon was to only consider user interface events that had a count value (across the entirety of the experiment) equal to or surpassing 10. In the chart, we can plainly see how in session 3 and 4, Result page events analysis directly precede Design Problem page events; this suggests that the designer used resulting data from

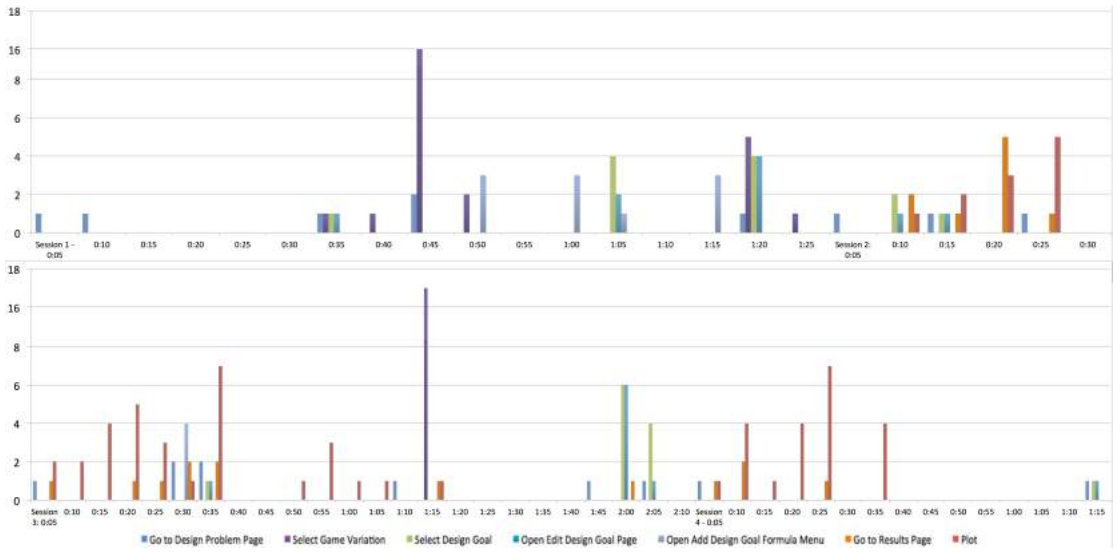

FIGURE 7.3: Sequential time-line with 5 minute intervals showing counts of all UI tool interactions (with a count value equal to or greater than 10), for Sessions 1 and 2 (top) and 3 and 4 (bottom). In blue and green tones are interface events related to defining the design problem. Several moments in these sessions did not have any interaction with AGE, and only involved the designer pondering and expressing his design and results with experimenters. In red and orange tones are events related to visualizing results from game-play testing. Note the vertical scale has been manipulated for issues of space.

play-test sessions as a basis for alteration of his design, engaging in an iterative cycle of evaluation and design.

We find high incidence of guided interactions with the Design Problem page at several moments. During a majority of Session 1 (all but before the 25 minute mark, which is when the designer was learning how to use the tool), the middle and ending of session 3 (0:30–0:35) and a small incidence at the very end of session 4 (at the 1:15 mark), so small in fact, it had not been clear in previous data so far. High degrees of interaction with the results page, indicating inspection of player-experience data, are found in session 2 (up to 0:25), first quarter of session 3 (until 1:05) and 4 (0:35). These moments, in session 3 and 4 particularly, seemed to have served as support for consequent design revisions (note how red blobs precede blue), whereas in session 2 the designer only looked at data (the reason for this is made clear in the next section).

To conclude, we list the result of the Creativity Support Index questionnaire filled by the designer at the end of Session 4 in figure 7.4. Note that, because we have only one subject, and no point of reference to compare it to, we cannot

use CSI to establish any form of validation of the tool. We only used it here as post-experiment survey that can help reveal the designer's perspective of the tool, and publish it as a comparison metric for future research (as per Cherry and Latulipe, 2014). That said, 89 out of a possible 100 points, paints a flattering picture of the tool. It means he found AGE for this design case a 'B+', or 'Very good' (according to the CSI's classification system).

CSI	Counts (1–5)	Score (0–20)	Weighted Score (0-100)
Results Worth Effort	4	19	76
Exploration	5	18	90
Collaboration	1	16	16
Immersion	0	14	0
Expressiveness	3	17	51
Enjoyment	2	17	34
CSI score	**89** (out of 100)		

FIGURE 7.4: CSI questionnaire results. Counts are the number of times each particular factor was prefered over the others in a Paired-Factor Comparison. Score is the sum of two user-attributed agreement scores (1–10 scale) to statements pertaining to that factor. Weighted Score is a ponderation of those two factors. And the CSI score quantifies how well the tool supported creativity for this user and particular task.

Perhaps more importantly, the CSI result highlights that the designer found AGE particularly useful in terms of two factors: 'Results Worth Effort' and 'Exploration'. The former means he judged the amount of effort required to use AGE appropriate, and the latter that he deemed it especially useful to consider different possibilities or alternatives and try out new ideas. We would not value the good rating in Effort as significant, as the designer did not implement the Base-game nor its changes throughout the process; therefore, his effort assessment is skewed and not representative of the actual effort needed to use this tool. As to the Exploration score, it reinforces the idea that he appropriated the tool only to experiment with the base-design, testing new alternatives (as the base-game edits attest to), and stuck to a divergent phase of creative exploration (never trying to converge on an intended design).

7.4 Design Walkthrough

In this section we analyse, step by step, the designers work, supported by his own verbal expressions and seeking to further interpret events.

7.4.1 Session 0 – A Design Brief

The designer's brief was expressed in a note written during the preparation phase:

- *"enemy 'attacks' give life - it's love.*
- *standing still* [you] *lose life.*[you need to]*"dance dance..."*
- *atacking the enemy kills him. "God is good but the Devil is also not bad." F.P.* [Fernando Pessoa, portuguese poet]
- *level* [goal] *condition is not to destroy the adversary."*

Later, in Session 1, he would come to say his design was meant to *"subvert the game a little"*. Whereas the base-game Dune is a strategy, militaristic game, defined by antagonistic confrontation between several factions, this was to become a game about life, love and dance. He wished for the player *"to understand that there is a different logic in the game, that there is a new exploration of movement (...) that wasn't implicit in the original design (...), and is more choreographic (...) so there is that main intention, and then there is the intention that the player understands and plays with the idea of receiving bullets backwards, or receiving life"*. His design assumed a number of alterations to the base-game:

- Enemy attacks increase player units health.
- Stationary player units lose health constantly (rate is once per second). Moving units are immune to this effect.
- Level goal was to collect a fixed amount of the spice resource (without perishing in the meantime).

In the first session he added his main design challenge: *"these were the three aspects [referring to the main three changes in the original game], these three aspects will come into conflict ... there is a dillema to manage there in terms of the very design and hopefully the system would help us create that, or find those, to balance that design."*

Also, he asked for set of new Game Parameters for the PCG algorithm to vary, because *"these things – how much life you lose while moving, how much you gain by being attacked – need to be balanced"*:

- How much health player units receive when attacked.
- How much damage to enemies' health player units deal.
- How much health player units lose while standing still.
- How much health player units lose while moving.

And finally, the designer required new data to be present in the game-play

metrics extraction system, referring to each units' status (whether moving, stationary, attacking, etc). This was needed to capture indicators that measured how player behaviour reacted to his design, and whether players learned how to solve the game's challenge. At this point he also expressed he wished to measure and (in the future) optimize three indicators: game time, number of enemy attacks, and player units exposition to enemy attacks. After this preparation phase, all these requirements were taken and implemented into the system.

7.4.2 Session 1 – Materializing the Brief in AGE

After preparation, a design phase with AGE ensued, in which the designer further defined his Design Problem. He started by creating Game Variations, and deciding each parameter range; when asked why he chose those, he replied *"I defined this with minima and maxima,* [referring to their boundaries] *thinking that their averages would be the values"* and then proceeded to point to the middle of a Game Variation range.

He proposed 3 Design Goals. One covered the duration of each game-play session, so to make the game *"relatively quick"*, he scored positively games that lasted around 5 minutes. Two, to have a per second average of player clicks on enemy units below 1, so as to signal the players compliance to not attacking the enemy. And three, to have the player dance, he wanted the average of movements per second to be as high as possible, so he made a score function that attributed higher scores as they were closer to 3000. So he wanted a game that had short play-time, where players did not attack the enemy frequently, and moved around a lot.

When asked if the player experience he wished to achieve would be totally in line with them, he replied *"not necessarily, might be down the middle, a 5 or 6* [in a 1–9 scale, 1 completely unaligned, 9 perfectly aligned]. *It is very difficult for me to be able to predict other results which I might not be contemplating"*. Despite this, when queried if he considered himself successful in translating his design with AGE, to which he replied *"impeccably"*. When queried to attach a number on a scale of 1–9, where 1 meant 'I Struggled Heavily' and 9 'I was perfectly capable of' translating my design, he replied with 8, which assured us he had found little difficulty in using the AGE application and ontology.

With the Design Problem fully defined, the designer deployed AGE to start play-testing; however, he did not set the system to optimize variables, only to

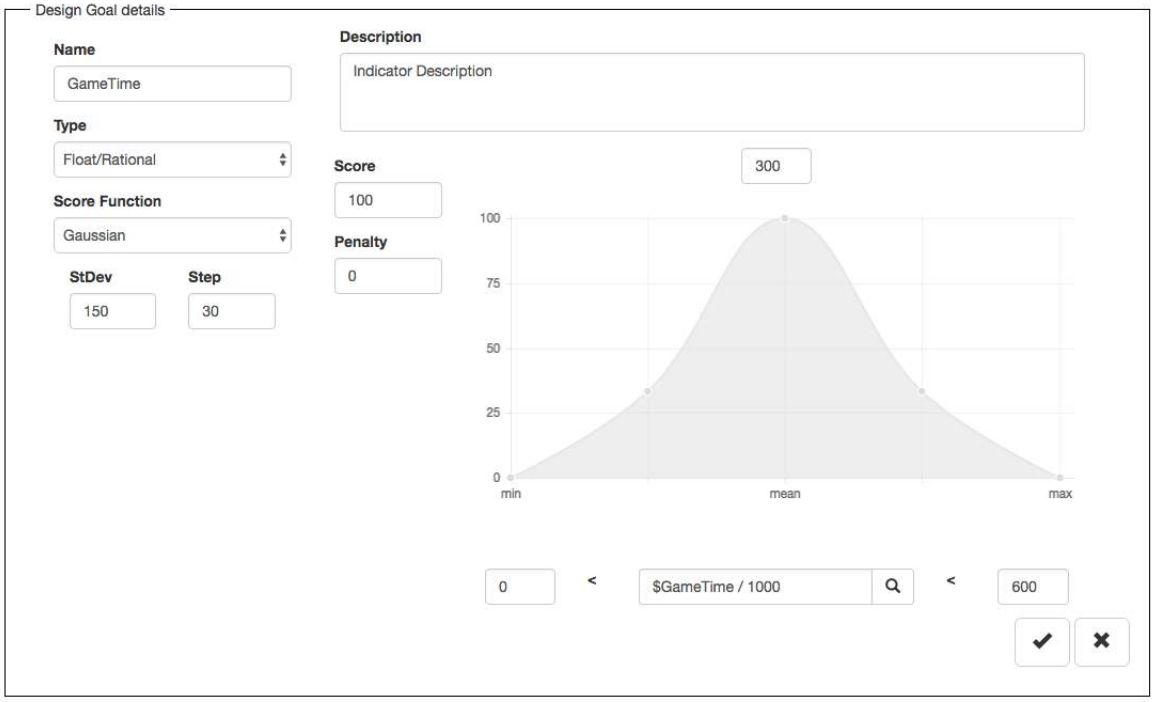

FIGURE 7.5: Screen-shot of one of the Designers' Design Goals. In this screen, users define how AGE should score and evaluate a given candidate Solution, and therefore optimizes generated variants. Users can edit an indicator formula, select boundaries of values which get positive scores, and even model the function that attributes the score.

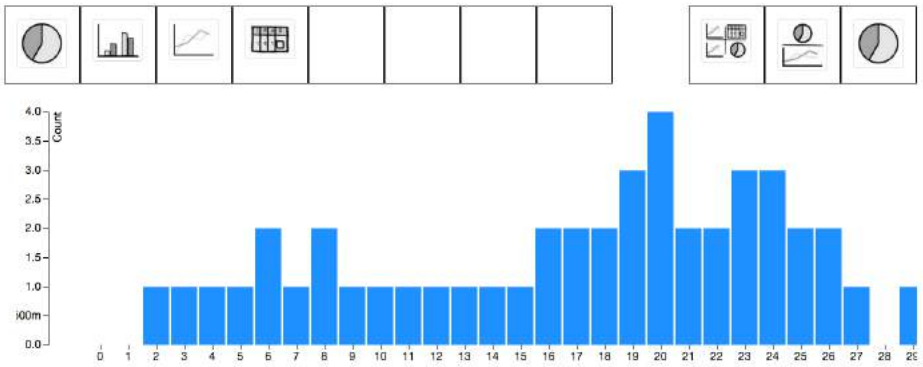

FIGURE 7.6: Screenshot of the results pane where Designers can inspect what occurred in playtesting sessions.

experiment. In this specific use case, the system uses random Game Parameter values, and tests Game Solutions using the Design Goal tests, but does not proceed to optimize. It works as a form of expanded AB testing that covers a great deal of random variants. As to the why of this choice, he explained that *"in this particular design case (...) we are starting with a pre-existing design and I am in a way manipulating that design, (and) as I do not have complete mastery of the pre-existing design, this experimentation will be useful"*[to understand it].

7.4.3 Session 2 – The Need for Better Data

Once a first batch of play-testing was done, the designer was invited to come and resume his interaction with AGE at his first convenience. After reacquainting himself with what he had done previously, he went to the results page, and inspected the values for the indicators he had used as basis for his Design Goal tests. Unfortunately, results were not in line with his expectations, *"This [a value of 7.3434242432e-8] is not very nice to analyse, considering the scale"*. *"People played considerably longer than we expected"*, leading to per-second averages of attack clicks and unit movements so low, that they neared zero for play-test sessions, rendering these indicators effectively useless.

To circumvent the problem, the designer proposed a two-prong approach: the addition of new charts for the AGE's results pane, and two new game-play metrics to be incorporated in the base-game's logging system, one to track the end-game status (whether players were winning or losing the game) and another to inspect how many player units were alive at any given moment (to provide a way to balance the assessment of number of player movements). One of the major points in his discourse, was that with the new charts and metrics he could

"measure the learning curve; for example, whether he understood [how to play the game] and when, when there is a convergence to a specific logic: more attack... less attack. I am still trying to figure out what can influence what (...) For this point, this initial point, I would like more to learn, to look at these charts in greater detail and understand. But maybe after two or three iterations, experiments" he would be *able to translate it* [the desired game-play pattern] *into an exact value"*. Thus, he deployed a new play-testing phase, to get data for the new metrics.

7.4.4 Session 3 – Clearer Picture of an Imperfect Design

With the new charts, the designer could better assess what was happening in the play-testing sessions. At this point, *"the player does not even have the opportunity to realize what is going on, because he is always winning the game"*. In summary, players could easily find a perfect strategy for winning, by accumulating the spice resource, moving along, while not minding the enemy, as its attacks gave life, and the life players lost while not moving was just not enough to hinder their progress. Alternatively, destroying the opponent – an easy task as they cannot harm players, led to an ambiguous win scenario, as despite not being clearly stated as a goal (players were told to collect X resources at the beginning), still had a mission success screen. The game is *"not too interesting, because if I wander from one place to the next, I win the game; it's not bad, but it isn't... cool"*. Also, he feared the message might not be coming across, for it lacked *"a dilemma, something a bit more moral"*.

Several different alternatives were elaborated aloud by the designer on how to solve this. Some were immediately discarded as he was not happy with them (these are the Cut Base-game Edits in figure 7.1), but in the end he was content with 3 changes. The first was that the pseudo-damage from enemies became the only way of getting the resources needed to win the game, *"instead of that* [pointing to the screen] *being the value of spice, it is a number that is incremented whenever the enemy attacks, which is love; what you would be doing is collecting love. (...) It is the irony of the material* [the Dune game] *and love; as players will have to spend quite some time realize how they can... gain love. Basically, the player is a bon vivant who likes dancing, dancing and receiving love"*.

Second, destroying the opponent would lead to defeat and be clearly stated as a lose scenario: *"there is this interesting condition that if I'm left alone in the game – this is related with the idea of love – I would lose the game"*. Third, the game would communicate 3 different messages to the player that would help him

interpret the new game rules; one when he starts playing, one when he wins, another when he loses. These messages – poetic quotes related to his game's theme – were forwarded by the designer, and each hints at what players must accomplish in the game. These, coupled with other minor alterations in terms of charts and metrics, were enough to warrant another play-test session.

7.4.5 Session 4 – New Design, New (Design) Issues

With the new design came new issues. While play-testers struggled at first to understand the game's new logic, in time, *"won almost everytime"*. Because players got resources and life from their opponents, they behaved like *"a leech"*. Though that meant his message was, at least partially, getting across, it still did not satisfy the designer, who started looking for ways of *"go about inscribing that [missing] tension"*, while maintaining the game's theme in the mechanics. After musing considerably, he changed the design problem via AGE, and decided that opponents' attacks should go back to removing player units life. This means that players would now have to juggle receiving damage from opponents, so as to gain resources, and moving – or "dancing" – with units to regenerate lost life, until the end-goal was met, while avoiding losing all units and destroying all the opponent units. He then deployed another round of play-testing, to see if the effect was as intended.

At this point, the allocated time frame for the test was exhausted. We then concluded the four design sessions without the designer terminating his process, as he was still in an active search for a game that was in accord with his brief.

7.4.6 Post-Experiment Interview

When queried on his difficulties in closing the design after four design sessions, he offered 3 explanations. The first was *"learning the tool itself, both the application's deal, its interaction"*; to him, this was *"a normal difficulty, that of learning [how to use] a new tool"*. We can, for the most part, consider this first issue as a necessary evil, as learning how to use a new tool is sure to require an investment in time and effort. This issue might be alleviated (through an improved interface), but can perhaps never be completely solved. Still, this has a a minor impact, given that designers would only have to learn how to use AGE once.

Most important, in his consideration, was a second factor: *"not knowing the [game] systems's behaviour (...) When I'm creating a new game I have a certain idea*

of what variables and system behaviours are at stake (...) and in this case there was the added difficulty of me not having it (...) this is profound, because it is not just a difficulty arising from my knowledge of this game in particular, but the game's genre. For previous games of his, he claimed he would be able define the Design Goal tests *"perfectly"*, for in his games, *"there are these minimum conditions for the experience to be favourable; favourable in the sense of aligned with your intention (...) and it is possible to define those tests"*. But for this case, he struggled. This resulted from an imperfect experimental design on our part, as in lieu of not demanding the designer to do the ground work in the development and integration of the base-game, we offered him an unknown base-game, one from a genre (real time strategy) which the designer was not comfortable with or knowledgeable of.

The third reason he gave, was that *"the type of design I tried to define (...), the type of experience, is hard to objectify with a set of metrics or values (...) it is hard to translate to a set of... to a mathematical model and say OK, this is the space I am looking for"*. So, in his mind, although he found the tool *"always useful"*, *"there are design exercises, game experiences (...) where you can extract much more value from this tool, than what we were trying to achieve"*.

Despite these issues, he was content with the direction of the procedures. The designer reiterated that AGE, even merely for exploration, was very useful, *"as [this design] implies a level of balancing which is not easy to achieve, and if it helps that process, it is impeccable!"* On a scale of 1 to 9, on how well he could use AGE to translate his design, to define the design-problem and design solution, he attributed an 8. Similarly, his prediction of how good the game would end up being was an *"8 or 9, because I've had the possibility of «seeing» people play the game and I think it is fulfilling the... [goals]"*.

Prompted to compare designing with AGE with his previous design processes, he offered several considerations, and opposed to our expectations, he said that, *"the great advantage I see in this solution is the structuring of the process, the guarantee (...) that using this process I can greatly increase the odds of finding a satisfactory solution. This possibility, of me following the process, and iteratively gaining feedback of (...) what is happening, in a structured systematized way, is the great advantage. And it is in this case that I think it will compensate the effort of configuration, measurement, programming, everything..."* Referring to a previous simulation game of his,*"because in games with strong simulation it is humanly impossible to test a vast array of possibilities"*, he said *"it would have been interesting to do that design with this tool in mind, to allow us to test several spaces of possibilities"*. Here

we note how optimization is not a keyword in his speech when discussing the advantages of the system, but the idea of a systematic process, of exploration of the design-space, and consequent experimentation and evaluation. Further confirming this alternate mindset, he mentioned that *"as we are using a tool that (...) computes the search-space, you're invited to mentally compute that same space; the way to configure the space that the tool offers, is also a way of rationalizing the process, and that helps"*.

7.5 Discussion

After the 5 sessions, the designer had not yet arrived at a point of trying to evolve his game, nor did he find the right game design that fulfilled his experience agenda; these were both unexpected results as the AGE tool was initially created to help designers improve mediated player experience. This may be a hint that there is an intrinsic problem with this approach. Either the focus on optimization is misplaced, and AGE is better suited for more experimental design procedures, or traditional design processes are better suited for creating a game that fulfils a specific player experience agenda. Of course, this might just be a constraint of this case study; conclusions must wait confirmation of further research.

The designer had no problem defining Game Variations; he was able to propose his design's search-space and stuck for the most part with it, as post-first design phase, there was only 1 editing event of a Game Variation. We think this signals that materialising the design's search-space was an intuitive process. However, despite early definition of Design Goals, he did not use them to the effect of evolving his base-game automatically. But in his discourse we find indicators that he had a clear mental picture of what behaviour he expected from players - to dance, to avoid killing the opponent, to offer himself to being "damaged" by the opponent. He did encounter a problem when trying defining concrete values: how much movement was enough? How much was too few? Equally crucial: what is a good metric to measure these somewhat abstract concepts? Twice, he went as far as playing with values in a calculator, in a visible effort that led to unfruitful results. Even his proposals for new data sources and charts did not solve this.

So, while the designer could verbally describe his desired form of player experience, he could not translate it into a set of final measurable conditions,

which is what AGE requires for optimization. His main answer to justify this, was that because he had not designed the base-game, he would always have to play around with it so as to understand it and get a feel of how it could mediate experience, thus proceeding to edit and analyse results from play-testing, until it would be sufficiently in line with his agenda. He never optimized, because this game design was not (until then) adequate to that end and because he had not yet fully grasped the existing game prototype and player-experience. But the uncertainty remains: can designers easily transport their vision of experience into a set of objective tests that automatically evaluate games? The designer could materialize these in abstract, just not in the concrete. Our answer to the question of how game designers *interface with procedural content generation algorithms so that they can solve game design problems?* is therefore still not perfect. The Design Problem metaphor, while still not perfect, seems to serve its function as a way for designers to interface with procedural approaches. Also of note is that the designer questioned the nature of this game and the moment in the design process in reference to their adequateness for experience optimization. Thus, **In what design contexts can experience optimization be used effectively?** is an open question for future research.

We do not consider these negative outcomes: the designer was clearly happy with how the design was evolving (both the CSI and post-experiment interview attest to that) only the process was slower to gaining momentum than expected. The reasons for this protracted development seem accurately described in his final interview, and we find no evidence to the contrary.

Additionally, we found several conclusions that we feel are of interest to the community from these results. The first is that **PCG can be used for Exploration**, meaning procedurally powered applications can be used not only for optimization of objects, but also for exploration of the design-space, in search of non-optimal configurations that can latter be iteratively improved. Three strands of evidence confirm this: 1, the designer himself in his interventions (especially the final interview), 2, the nature of the design process, where PCG was used only for exploration purposes, and 3, the CSI questionnaire, where the highest score was in the exploration category. This was a somewhat unexpected value found in a tool that uses PCG, as there is a great deal of research focused on optimization. Results suggest effort should be put forth on how PCG can be used for divergent exploration of the design space by authors.

Furthermore, the tool's systematic approach to game prototype deployment

and data collection (a semi-automatic emulation of traditional design processes) and the accompanying mindset it suggested, was valued by the designer in of themselves, despite any possible gains in effort permitted by PCG. Then, one can perhaps conclude that PCG was not even the greatest value addition in the tool. Hence, **Tools that systematize prototype development, evaluation and testing processes can aid game designers**. This shows that our desire to automate early cycles in the game-design and development process is valid, and that AGE – with or without PCG algorithms – assists in that direction, providing a valuable backbone to the design process. Its imposing structure – an iterative cycle of 3 phases: design problem definition, play-testing and experience data evaluation – seems powerful enough to aid game designers.

Despite focus on exploration, the designer claimed that in the context of his past game designs, he could see himself using AGE to optimize them, assuring us that there is potential usefulness in that use-case. This reinforces the idea that there is a tendency for a phase of exploration of the design-space, before optimization/evolution of existing games. When asked on how he would, hypothetically, address other design cases – if he would use exploration as he did in this case – he replied that he would, *"normally (...) initially"* explore, but he also made clear that in *"other contexts (...) it would be nice to optimize"*. So we can at least hypothesize that AGE and other similar approaches should offer both, and that it is likely that **Exploration precedes Optimization**.

Even though the designer struggled with defining the Design Problem with it, in time, he was able to understand all the terminology, metaphors and procedural apparatus. The design process he embarked upon shows all the signs of early, creative exploration work: he designed a game, collected data, redesigned the game, and could surely do so continuously until he found a solution. AGE assisted by providing an easier way to explore different design regions simultaneously, a structured work flow, and an application that eased the process of player experience data collection and analysis. Despite the issues, its appropriation by a designer in a creative design process is a moderate success, showing the approach's feasibility for assisting designers.

Data from these case-study sessions attest to AGE's potential utility to designers, although mostly in an unforeseen light. This being only a case-study means it serves only to give tentative answers that empirical studies can later confirm. With this, we conclude the collection and analysis of data that allowed us to start answering our 2nd major Research Question, *"How does the introduction*

of AGE transform game design processes?". The next chapter will conclude this thesis, reviewing all the answers to the research questions, and listing all of this research's contributions.

Chapter 8

Conclusions

This chapter closes this stage of this research project by summarizing our findings, detailing the breakthroughs and limitations that we found, and by pointing out guidelines for future work.

8.1 Research Questions and Answers

The first set of research questions pertained to the tool's design,

1. *How can game designers use experience evaluation methods and procedural content generation algorithms to prototype video games that mediate intended forms of player experience?*

The proposed AGE's model, as specified in this document (and further detailed in both design documents and instantiations) is an answer to this general research question. In other words, the answer lies in finding tools that enable the use of player experience optimization methods by game designers, in ways that afford them fine control over which target forms of player experience to optimize towards. We consider that the challenge in conceiving these tools has little to do with the technological limitations of advanced problem solving algorithms – as the literature review showed, powerful methods already exist. As we showed in the Mario experiment (section 5.3), an out-of-the-box general purpose search algorithm can solve a simple design problem. In our perspective, it is not for lack of powerful algorithms that these techniques are not more wide spread in creative practices. We do not lack powerful procedural algorithms; we lack powerful control interfaces. Interfaces that can open these complex methods to non-specialist designers – this is the main point of AGE and its design. As we delivered a working interface to these search algorithms, we established

a method for designers to optimize intended forms of player experience. The Mario experiment shows that this approach can serve to optimize player experience, and the Case Study demonstrates that designers can (to a point) use this approach for their practice, assuring us that AGE's model is an answer to this research question.

(a) *How can we delegate parts of the game design process to automatic generation algorithms, in ways that enhance designers creative authorial control over the resulting artefact?*

Our approach to solving this question is to have automatic generation algorithms manipulate material qualities of the end-game artefact constrained by two major factors. First, have designers in as much control of the generation process as possible: allowing them to choose which material qualities to forfeit control of, to define the boundaries of the search process, to define some of the mechanical aspects of the search algorithm and to select which algorithm to utilize. In summary: allow designers every strategy that increases authorial agency over the automatic generation process. Second, to have designers determine what is the experiential end-goal towards which material qualities will be manipulated by automatic generation procedures, and therefore giving designers higher authorial influence over the form player-experience ends up taking. Hence, designers are turned into a form of second-order designers, or meta-designers, which can model player-experience through the AGE tool interface (or some other AGE-like approach). Together, these two qualities distinguish AGE from similar approaches, by clearly focusing on designer empowerment. And by doing so without requiring of them expert knowledge on complex Artificial Intelligence methods; the algorithms serve the user, and not the other way around.

At the moment, th

(b) *And what are the requirements for procedural content generation strategies so they can solve game design problems in this way?*

Though we have no firm collection of evidence to back our answers to this question, we have amassed some data (in chapter 5, and prominently in section 5.2.5) that points to a set of guidelines we can synthesize here. For PCG to be used in a manner envisioned by the AGE approach, algorithms should not make any major assumptions in regards to search-space composition, and be as resilient as possible to dynamic, non-differentiable, craggy search-spaces. To alleviate these conditions, we suggest restraining procedures to simple design problems

(meaning small number of variations and goals), using high numbers of playtesting subjects, and having them play candidate solutions repeatedly and in a randomized order. These strategies should allow for some statistical stability of experience indicators, and therefore, a smoother, more stable landscape for algorithms to operate on top of. This does come at a cost, as greater number of evaluations requires a greater number of players investing time in the playtesting procedures.

(c) *How can we combine experience evaluation methods with procedural content generation tools, in a process that can promote designer influence over player experience?*

In this regard, AGE answers these questions by allowing designers control over which experience evaluation methods to use when assessing candidate solution's fitness, and how they do so. Designers provide AGE with a model (in the form of design goal tests) with which to evaluate each candidate solution in terms of player experience. Data from player experience is not taken as is – it is processed by AGE using designer guidelines. Hence, by adopting a Quality Definition strategy in terms of how Designers can impact PCG evaluation procedures, we harness the value of automatic experience evaluation methods (such as game-play metrics, bio-metrics and self-report questionnaires), without losing designers capability to influence the end-result of the process.

Granted, at this time, we do not have any empirical data to support the relative adequateness of this solution face alternatives, but we do have accumulated enough data in our experimentation to know that this strategy is feasible in terms of problem solving (we know that from the Mario Experiment), and two, that designers were able to understand this approach and can, potentially, use it to explore alternatives (as seen in the case study).

(d) *How can game designers interface with procedural content generation algorithms so that they can solve game design problems?*

This question embodies one of the greatest challenges in this research process: finding the right interface and accompanying terminology that allowed for designer appropriation of the AGE tool and approach. Briefly stated, the best results we have found are structured around a test metaphor – have the procedural content generation strategies evaluate candidate solutions on the basis of an if-then-else rules structure. If a certain experience indicator has a given value, then attribute it a score of X and if it does not, attribute it a penalty

of Y. This simple logic allows designers to craft complex evaluation functions with a relatively simple design vocabulary, and in doing so, finely tuning their desired player experience with procedural content generation algorithms. The Case Study showed that a designer can state his intent for player experience using our Design Problem metaphor; however, he did not optimize game solutions using his Design Goal tests, so the answer is still tentative. It seems this format we created is sound enough that a designer can express his design based on it, but we cannot confirm its degree of operativeness.

Another crucial aspect we found in respects to the interface's form, at least in reference to the audience we tested the interface with, is that clarity and simplicity were more important than advanced control. The more problematic user interface issues we discovered have to do with understanding advanced functionalities of AGE. During participatory design sessions a significant part of the debate concerned how to make the process intuitive and fast. Striving for interface design simplicity and process bootstrapping workarounds is crucial. Dealing with complex procedural approaches is only possible if the interface eases designers into this convoluted milieu. AGE's interface already goes a long way of bridging this gap, yet is far from perfect. Future design revisions must make further efforts in tuning the terminology, the design metaphor, and the way the interface communicates the process.

As an indirect response to this question, our research also led to the development of the Actor-role Configuration Taxonomy. It serves as an expanded response to this question, listing all possible ways in which designers can interact with PCG approaches in game design processes, even beyond the context of the AGE tool.

The second set of questions relates to studying the impact the AGE solution has in the game design and development activities. Meaning, how well does this tool solve the problem, and what effects, both positive and negative, does it have in the process it aims to improve?

2. *How does the introduction of AGE transform game design processes?*

The AGE design case study serves as an illustration of how AGE changes game design processes. By establishing a structured, semi-automated process of game deployment and player experience evaluation, it pulls designers into iterative cycles of design problem definition and result evaluation. Not only does it structure the design process, it explicitly functionalizes it as a form of player

experience modelling (by way of focusing procedures on indicator assessment and testing). In our illustrative case-study, the designer found this structuring useful and valuable for his design activity. Moreover, this is the exact same focus that game design and production experts and insiders consider essential for good results (Schell, 2008; Fullerton, Swain, and Hoffman, 2008; Brathwaite and Schreiber, 2009). While this convergence does not make for a transformative change face designers non AGE-mediated practices, it gently pushes them to what are consensually viewed as best practices.

Additionally, AGE's contextualization of automated generation procedures opens designers up to new creative possibilities. In theory, the optimization of a given prototype can allow designers to more finely tune their target player experience, finding a more adequate solution to their design problems, and providing them with immediate player-driven validation of that same solution. And as seen in the case study, a designer can also take an exploratory approach to problem solving, and simply wander the design space, assessing the mediated experience of local variations of a given prototype. Both these approaches – optimization and exploration – are semi-automated. Therefore (in theory, at least) diminishing their cost of use and increasing designers capacity to engage in these tasks given a fixed (time, cost) budget.

(a) *With what processes and goals can game designers appropriate AGE?*

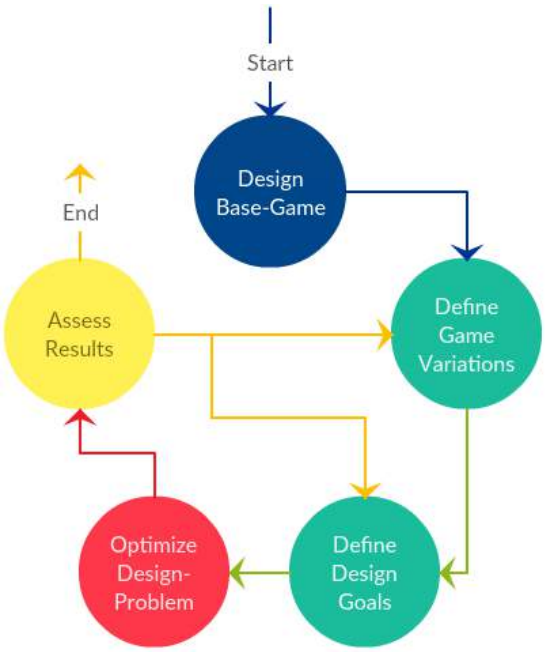

FIGURE 8.1: AGE's Proposed Task Flow

The AGE proposal was designed from the ground up to support design-time optimization of game content (figure 8.1 shows the implicit task flow in the AGE proposal). This, in itself, is a radical shift face the more popular and widespread goal of PCG, which is to generate large quantities of content with a lower production cost, either during design-time (to increase shipped content) or after shipping, to increase re-playability. By putting the focus on player-experience optimization, AGE's goal differs widely from such alternatives, with its emphasis on improving the creation of higher quality content as opposed to higher quantity. If one views the betterment of player experience, this change in goal seems as valid as others, and has the advantage of utilizing PCG in a way that protects player consumers from contact with PCG's common pitfalls (access to repetitive, uncreative content). However, exploration is the use-case we were actually able to observe in the case study.

Though designers in participatory design sessions and the case study were always interested in AGE's intended operation mode, the fact remains that we have little empirical evidence to back the optimization use-case for AGE. Discussion in these sessions (let-alone all the theoretical background that sustains its design) point to the optimization use-case being valuable and we are confident that given the right design context AGE can be used in that manner. It is plausible, and confirmed by designer himself, that the case study simply was not the ideal design problem, at the ideal design stage, to use such an optimization approach.

From the participatory design sessions and from the case study we have found that designers would want to appropriate AGE with the mindset to explore the design-space and experiment with a given prototype's variations (figure 8.2). This allows designers to accurately map out how given game parameters affect player experience, therefore enriching knowledge of their object's effective mediation. On top of this, in some contexts this could also ease serendipitous processes of discovery of new game configurations. Optimization allows for the creation of a high-quality variation of a given game; and Exploration complements it, by allowing the search for interesting, novel and unexpected alternatives to existing designs.

(b) *What trade-offs result from substituting manual game authoring processes with semi-automatic processes supported by AGE?*

To use AGE, designers must develop a prototype base-game, and integrate it with the procedural generation and experience evaluation engines in the AGE

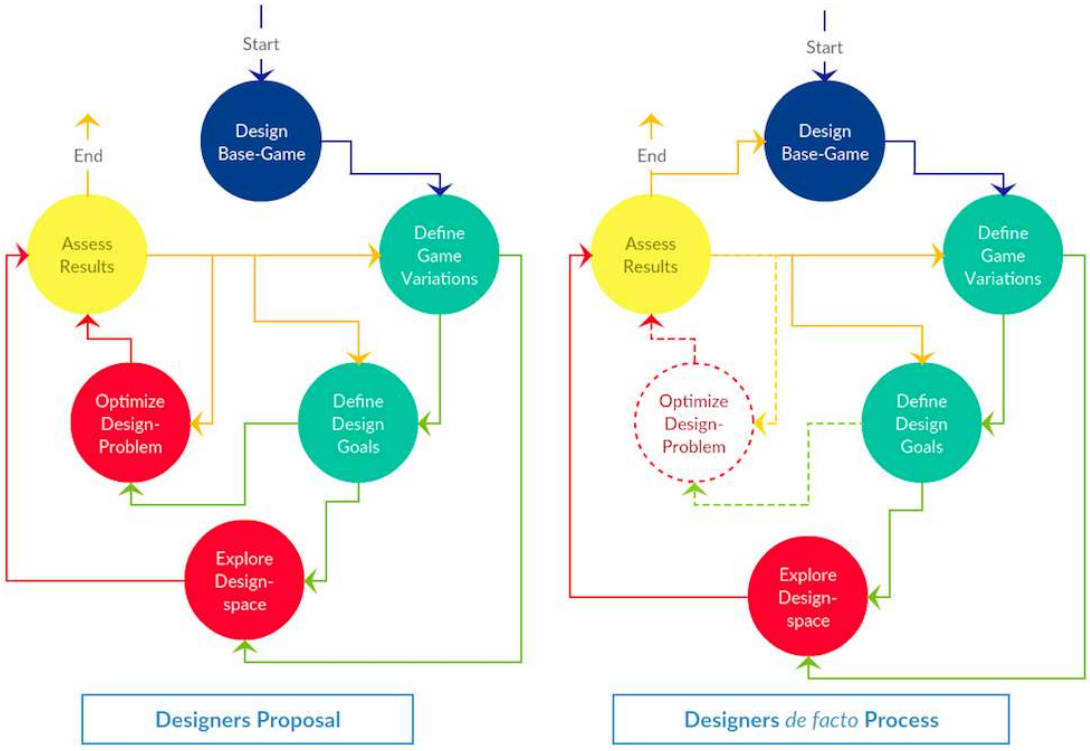

FIGURE 8.2: On the left, task-flow proposed by designers in the participatory sessions. On the right, the case-study designers' task-flow (optimization is left dashed as the designer declared he would use it in different projects, though he never acted it out in the experiment).

platform; part of the integration involves coding specific routines inside the confines of the base-game. Naturally, this is a time-consuming effort, besides requiring understanding of AGE's software architecture. The case-study designer felt this trade-off was worth it, even just for the sake of having a structured and instrumented game design process; PCG-driven Exploration and Optimization possibilities were an added bonus. While his view is positive, one must keep in mind that he did none of the required ground work to get AGE running – his opinion must be read as an expert's speculative answer and not a definitive one. Still, considering the degree of effort that can be saved in having a highly instrumented design process and a tool that allows semi-automated improvement of the created artefact, this seems a promising compromise.

Besides the effort on the programming and design side, to use an AGE-like tool, production teams would need create a context in which several players play-test different candidate solutions repeatedly, and this will likely come at additional resource cost. This cost is further increased by the partial randomness of search procedures, which is likely to waste play-test resources in evaluating

less promising candidates. Additionally, as optimization procedures will provide better results if play-test data is statistically well behaved, there are instances where this cost will not be worth it, and where manual, designer-oriented iteration of solutions should afford a better cost-benefit relationship. What this also signals is that, because AGE is a tool so dependent on human designers, it ends up being up to designers to discover which design problems are beneficial to solve with it, and how. Selecting the right combination of Game variations and Design Goals, the right play-tester sample, the right strategy – either Exploration, Optimization, or just simple testing of a single candidate – and the right phase of the game design cycle, become crucial skills for designing with AGE.

As an offset to the potential increase in cost, AGE suggests greater participatory involvement from players in the design process at all stages. Designers will no longer need to make speculative assumptions on how players play a given prototype, nor will they need to actively engage in testing out what new configurations are needed to achieve a desired experiential effect – data from play-testing and procedural algorithms will take on (at least part of) that burden, all while providing greater predictability to the outcome of a design project.

(c) *Can the AGE approach foster creative processes in game design?*

During the case-study experiment, the designer gave ample evidence of being involved in a creative process. His design agenda seems creative to us, subverting genre expectations and the very point of the base-game's militarist logic and game design. Also, his design process was dominated by a care-free, explorative attitude that valued experimentation and serendipitous discoveries. When self-assessing the creative potential of the design process mediated by the AGE tool, the designer gave extremely high marks. And yet, he did not finish the design process – and this forcefully tempers our conclusions. We cannot assess the quality of the designed artefact because of this very reason. Still, this presents a powerful case study in advocating the possibility that AGE, at the very least, does not disrupt creative game design processes; instead it can be fruitfully framed in them.

8.2 Contributions, Constraints

In line with Vaishnavi and Kuechler (2004) stance on the value of Design Science Research, AGE's novelty lies in its design. Therefore we think it stands, by itself, as a valid contribution to both the game design and game research

communities, and as far as we are aware, it has no equal. Furthermore, through this research's empirical studies we have found evidence to support that: a) AGE can be used to solve design problems, even if merely paired with a simple search algorithm (as seen in section 5.3); b) a user can use AGE to model a real-life design problem (as chapter 7.1 shows). This means that AGE *can* be used to achieve the goal it was envisioned for. Our added emphasis on *can* (as opposed to will), is meant to recognize the fact that we did not establish sufficient grounds for generalization, nor can we lay claim to guarantee its usefulness for every design case. In this regard, it stands as a work in progress.

This thesis admittedly leaves several lingering issues open for further research. First and foremost, we did not demonstrate clearly how useful AGE can be to designers. The case study we presented shows AGE is usable and potentially interesting as a creative tool – by showing an illustrative case-study of a designers' creative work with AGE – but does not show a designer finishing a complete design process and delivering the end-product. Nor were we able to repeat this experiment a significant number of times to assess if it is frequently and consistently useful. Equally, experiments only addressed two fully developed game prototypes; questions remain on whether AGE is applicable to different games, different genres, and different phases of design and development cycles. Given the context and the resources available to us, as well as the highly experimental character of the tool's design, it was perhaps naive to expect all these to have come to fruition in the limited time-span afforded for this PhD. Further investment in testing AGE with design cases, especially scenarios where optimization becomes a use case, would be needed for added insight.

Against our expectations, this research was not seen as sufficiently meaningful in terms of its contributions to the Procedural Content Generation and Computational Creativity research communities. Whether by virtue of our Human Computer Interaction-focused positioning or our disinterest in advancing the algorithmic aspects of procedural content generation methods, dialogue with these areas was not as fruitful as desirable. A strict Human-Computer Interaction positioning was also far from ideal when contextualizing ourselves in the area. Not only was the topic too far left-field for being aligned with current community's interests, as the employed research methodology was perceived as lacking the expected investment in user studies and consequent generalization. In hindsight, we realize that ventures into this research's themes might have been more successful if we had started from within one of these core communities

and their shared interests before adopting an interdisciplinary approach.

We can also recognize that this research was framed in a nascent field that evolved rapidly in wildly divergent paths. This turbulent change is partly why we contributed with a PCG Taxonomy that can both help classify and organize systems, and point out interesting research avenues for upcoming ones. AGE's design itself, while extremely novel at its 2010 inception, has since been joined by other ground-breaking work (as seen in our literature review, section 2.2.3) that in many ways complements it. Other more recent ventures are likely already extending this lot, and providing even greater strides. Nonetheless, the unique character of AGE – its design goals, process, and philosophy – still stands as a model for a general-purpose design-problem solving tool for game designer use.

As an ongoing research project, AGE is open to future novel implementations, and can be appropriated and iterated upon by any practitioner or researcher that decides to do so. We are confident that the existing design can be vastly iterated upon, and that these first steps are worthy of ensuing research. Moving forward, a greater integration with both design and industry production contexts will be obligatory, with design and evaluation procedures moving from the laboratory out into the wild.

Finally, the case-study we covered in the final chapter illustrates a designer working with AGE in a novel use-case – using PCG for exploration of the research space. Such a shift in focus seems particularly interesting to us, and seems to point to new areas of research in adjacent fields. The usefulness of this exploration approach is subject to future research.

Design Science Research projects, to aim for meaningful scientific contributions, must not be exhausted in artefact instantiation; instead, they must strive for the proposal of abstract artefacts that can be generalized to other contexts (Gregor and Hevner, 2013). The AGE model, its design principles, as well as its admittedly imperfect empirical support, can be appropriated into novel scenarios and contexts. Furthermore, data from the four experimental chapters of this thesis, as well as the theoretical backgrounds behind AGE's design – especially the Taxonomical proposal that served to guide its positioning – point to several new Research Questions that future projects can study.

8.2.1 Future Work – New Research Questions

Future work must build on the hereto laid foundations. New experiments and new data are needed to continue answering the original research questions, and bring further validation of AGE's applicability and usefulness to design contexts. It is mandatory to test the approach in new case studies, with new prototypes, and new design agendas; these will serve to assess the tool's problem-independence, as well as validate its usability. This notwithstanding, as a corollary to the results we have insofar found (especially from data from the Case Study), a set of new research questions emerged, and that future research should strive to answer:

1. *How much value is there in simply systematizing game deployment and evaluation?*
2. *In what design contexts can experience optimization be used effectively?*
3. *How can Exploration become a valuable use-case for this (kind of) tool?*
   (a) *Is Exploration a task that tends to precede, or alternate with, Optimization in the design process?*

Besides this, once AGE has been more firmly empirically validated, we see the way forward by questioning:

4. *Can an analogous approach to AGE be used in other Human-Computer Interaction contexts?*

This last question seems imperative given the immense transformative potential in optimizing user experience for all families of digital applications. In a world where all human activities are mediated in some shape or form by computers and where their interfaces can impact so much of our lives, tools that can help move these beyond functional adequateness and into experiential adequateness are needed more than ever. AGE is a potentially useful model for that much needed radical shift from material design into experiential design (McCarthy and Wright, 2004). In this sense, AGE can become a contribution not just to the game design community, but to all HCI communities.

8.2.2 Publications

As a resulting product of this research, a number of scientific publications were published. What follows is the list and a reference to how they relate to this thesis corpus.

Parts of our literature review (chapter 2) were published in the following papers:

- Rui Craveirinha and Licinio Roque (2010). "Looking for the Heart of Interactive Media - Reflections on Video Games' Emotional Expression". In: *Fun and Games*. Leuven, Belgium
- Rui Craveirinha and Licinio Roque (2011). "Zero Lecture in Game Design". In: *Proceedings of SBGames 2011 Arts and Design Track - Full Papers*. Salvador, Brazil

The taxonomical model detailed in chapter 4 was published as:

- Rui Craveirinha, Nuno Barreto, and Licinio Roque (2016). "Towards a Taxonomy for the Clarification of PCG Actors' Roles". In: *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play*. CHI PLAY '16. Austin, Texas, USA: ACM, pp. 244–253

The AGE design proposal was published in:

- Rui Craveirinha and Licinio Roque (2015a). "Designing Games with Procedural Content Generation: An Authorial Approach". In: *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*. CHI EA '15. Seoul, Republic of Korea: ACM, pp. 1199–1204

And last but not least, three empirical studies – the functional prototype experiment (sections 5.3), the participatory design sessions (6.1.2 and 6.1.3 and their analysis, and the AGE case-study (chapter 7):

- Rui Craveirinha, Lucas Santos, and Licínio Roque (2013). "An Author-Centric Approach to Procedural Content Generation". In: *Advances in Computer Entertainment: 10th International Conference, ACE 2013, Boekelo, The Netherlands, November 12-15, 2013. Proceedings*. Ed. by Dennis Reidsma, Haruhiro Katayose, and Anton Nijholt. Springer International Publishing, pp. 14–28
- Rui Craveirinha and Licinio Roque (2015b). "Studying an Author-Oriented Approach to Procedural Content Generation through Participatory Design". In: *Entertainment Computing - ICEC 2015, 14th International Conference Proceedings*, pp. 383–390
- Rui Craveirinha, and Licinio Roque (2016). "Exploring the Design-Space: The Authorial Game Evolution Tool Case-Study". In: *Proceedings of the 2016 Advances in Computer Entertainment Conference*. Osaka, Japan: ACM

8.3 Final Remarks

The main purpose of this thesis was the design of a new way of framing PCG methods in a tool that could assist game designers in crafting high-quality video game experiences. The Authorial Game Evolution tool, its design and underlying step-by-step approach, are the solution to this problem. By constructing a prototype of AGE by way of Design Science Research, we have provided what Hevner et al. (2004) call *"proof by construction"*: in instancing a system that partially automates a design process, we show that this process can indeed occur in this semi-automated fashion. AGE allows users to define game design problems, and then maps these to (potentially) computationally solvable optimization problems.

Using the terminology proposed by (Gregor and Jones, 2007), our contributions to the community, are the AGE instantiation, the abstract design model used to create it, and the methods with which designers can use it. Though the design has not been thoroughly vetted, and though the AGE model still is not enough to present a stable scientific theory, the ground work here presented is a steady first step towards that end... one filled with possibilities for evolution.

Appendix A

Design Materials

This Appendix presents an overview of the interface prototypes that were designed and developed for AGE. It is a photo-album with illustrative snapshots of AGE's design-process, showing examples of the output of each step (note this is by no means an exhaustive look at the prototypes). The interface design process itself is further detailed in chapter 6. For each step in the design process there was a prototype designed either by the participants in the Participatory Design sessions, or by the thesis author. Results were iterated upon, meaning each prototype was an evolution of the previous one, in sequence (see figure A.1). Though changes had to be made throughout the process by the author, the application requirements followed first and foremost the designs forwarded by the participants in the PD sessions, and their expressed needs.

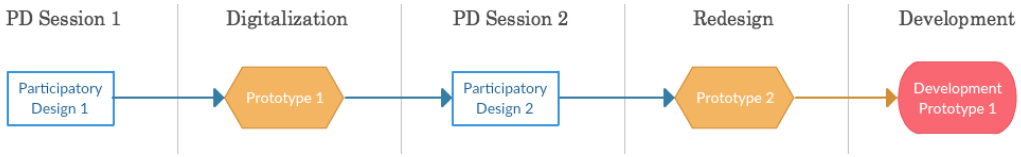

FIGURE A.1: Sequence showing when each prototype was created. In blue are the Participatory Designed versions, and in remaining colours the ones authored by the author.

The first prototype — Participatory Design 1 – was a paper prototype drawn during the first Participatory Design session. It served as a basis for a series of mock-ups by the thesis author, which gave it a more stable design and introduced basic AGE functionalities that had not been taken into account in the participatory session. The latter was then given back to the participatory designers for confirmation that it was as in line with what they designed; the result of this is Prototype 1.

A second Participatory Design session by a new lot of participants followed: edits made in that session to Prototype 1 are entitled Participatory Design 2.

From that, a new set of mock-ups – Prototype 2 – were designed; they introduced several changes to Participatory Design 2 as a way to solve lingering issues from the participatory design sessions. Prototype 2 served as the basis for a fully functional web-prototype – Development Prototype 1 – that remains in development as of now. The latter can be accessed on-line through the link crowdplay.dei.uc.pt/login. If you wish to survey the prototype, use the user name 'asd@asd.asd' and password 'asd' for the Experiment 'firstdemo'. What follows are photos of each design with some minor comments on changes between versions, to help contextualize these screens.

A.1 Participatory Design 1

This prototype was the result of the first Participatory Design session. It is comprised of four distinct screens: one for editing Game Features, one for Editing Game Metrics (for some reason, designers used the term metrics as if it were equivalent to experience indicators in general), an Overview pane which shows both Features and Metrics, and a Results pane. These would be the base four screens that compose the backbone of the AGE design, and that would remain relatively stable throughout iterations.

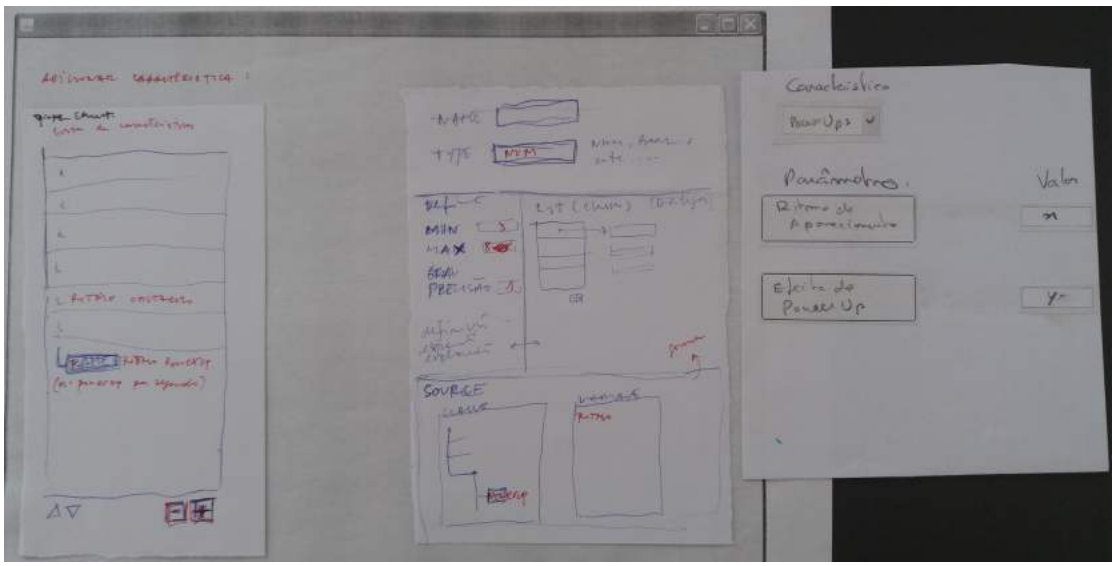

FIGURE A.2: Feature creation and editing screen. The vertical two pane layout would become a motif for all the designs, disseminated throughout most screens for consistency.

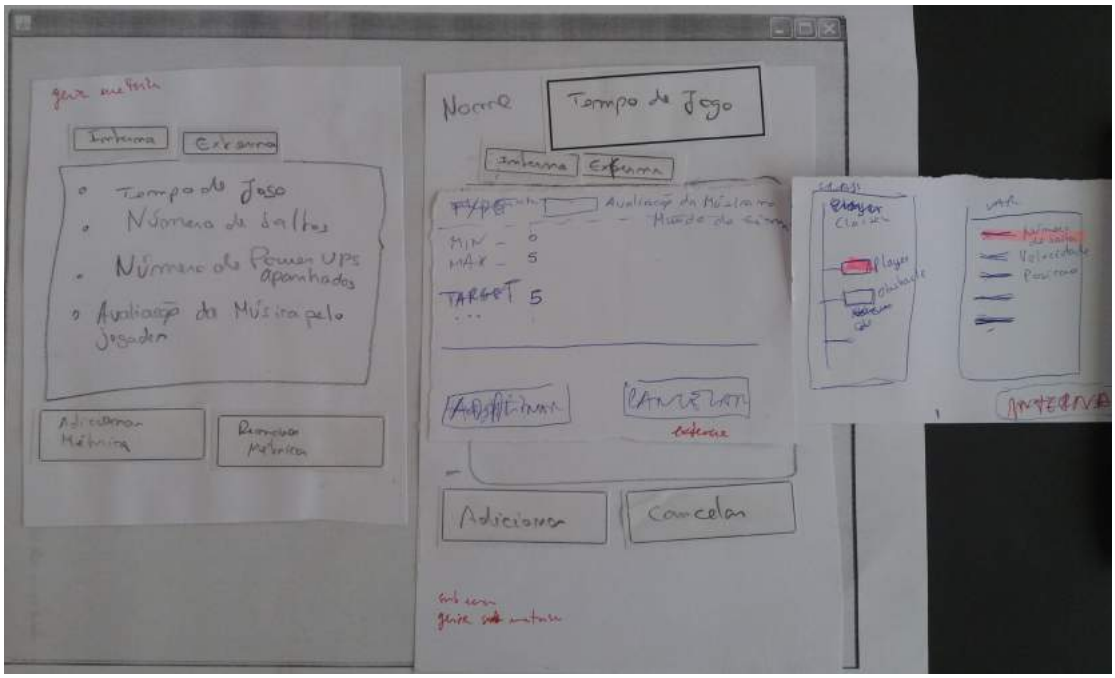

FIGURE A.3: Metric creation and editing screen. Designers used the term metrics in their design, but their speech was in reference to indicators. The basic outline of both Features and Metrics screens is a side by side view of a list pane and an editing pane. The list pane briefly outlines Features and Metrics basic information, and the editing pane allows editing of an entity's values.

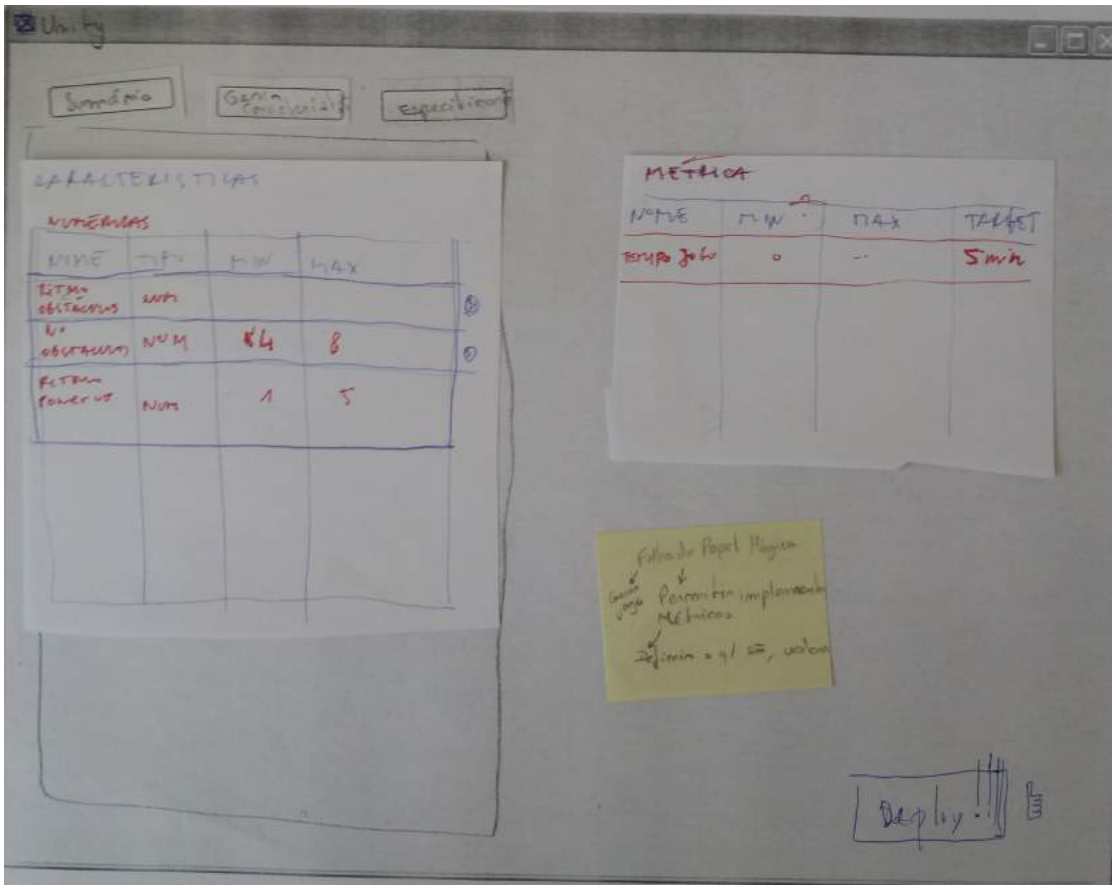

FIGURE A.4: Overview screen of Features and Metrics. The overview screen is a simple coalescing of the list panes of the Features and Metrics screen. For the designers, this allowed a general overview of the Design Problem that was to be solved by AGE. In later revisions, this idea of a basic Design Problem unit would be made explicit and further materialized in the design, precisely because of this screen.

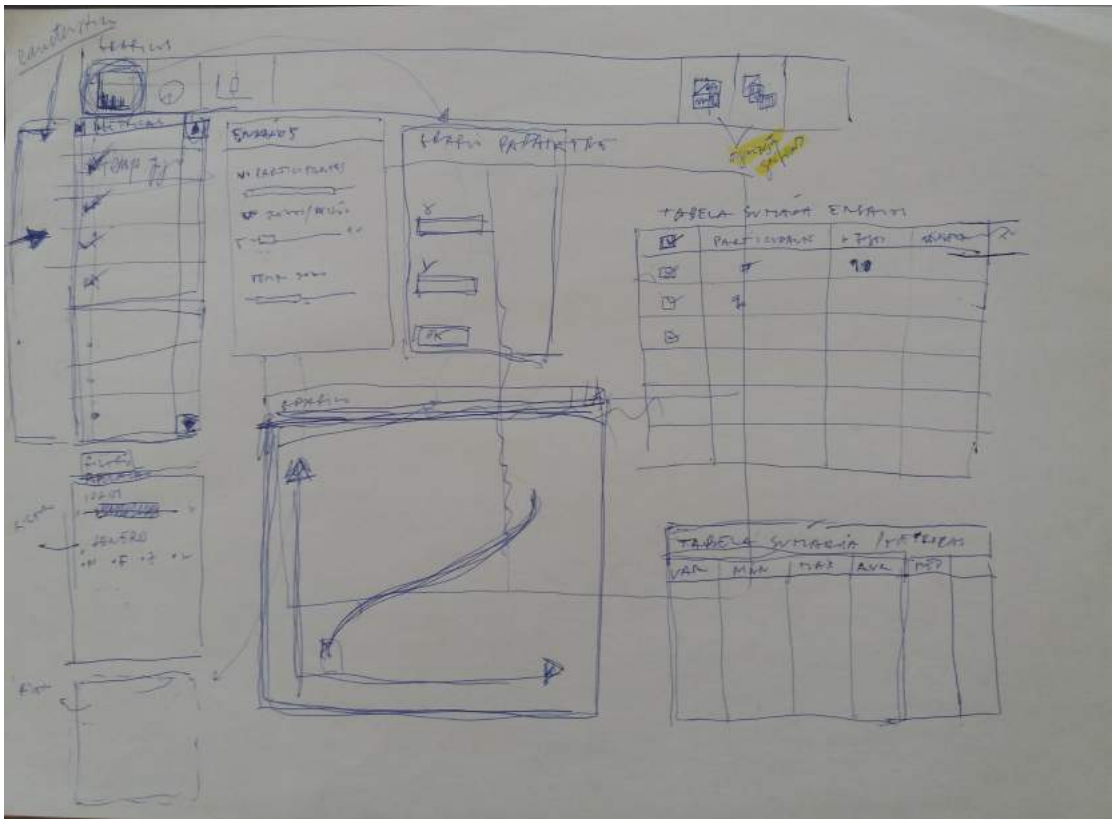

FIGURE A.5: The Results Screen. The results screen was one of the more surprising: it was extremely detailed compared to the others and showed a great deal of functionalities, such as multiple plotting panes, graphic types and a data filtering system. This is curious as AGE was originally intended for optimization, and given that fact, the need for complex data visualization tools should be lessened. Instead, it is a major focus of the design.

A.2 Prototype 1

Because the previous design was very rough, lacking in detail, and was made without taking some of AGE's functional considerations in to mind, a digital prototype was made based on it. Some minor additions were made to complement the original, while maintaining the layout and designer intent of the original. It should be noted that changes were vetted by the designers of PD1 before their inclusion.

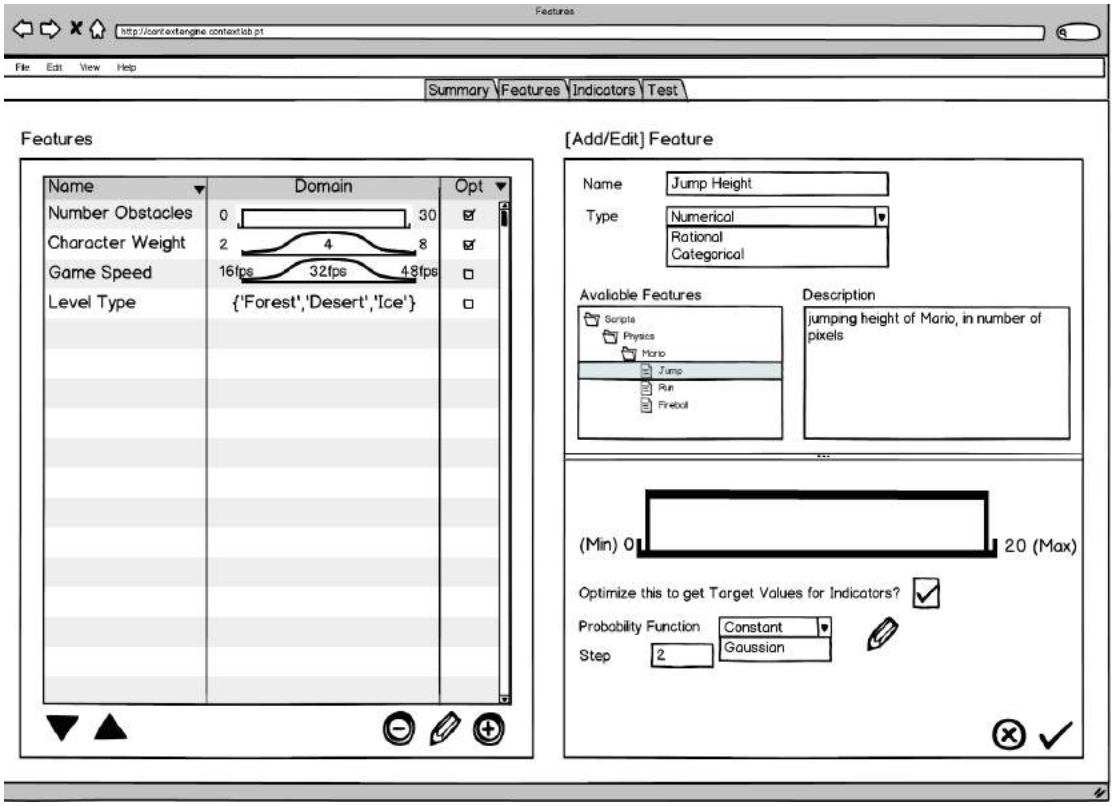

FIGURE A.6: Features Screen. The one notable addition in the Features screen is the non-mandatory probability function configuration sub-panel; this was meant add further authorial control over the procedural generation process.

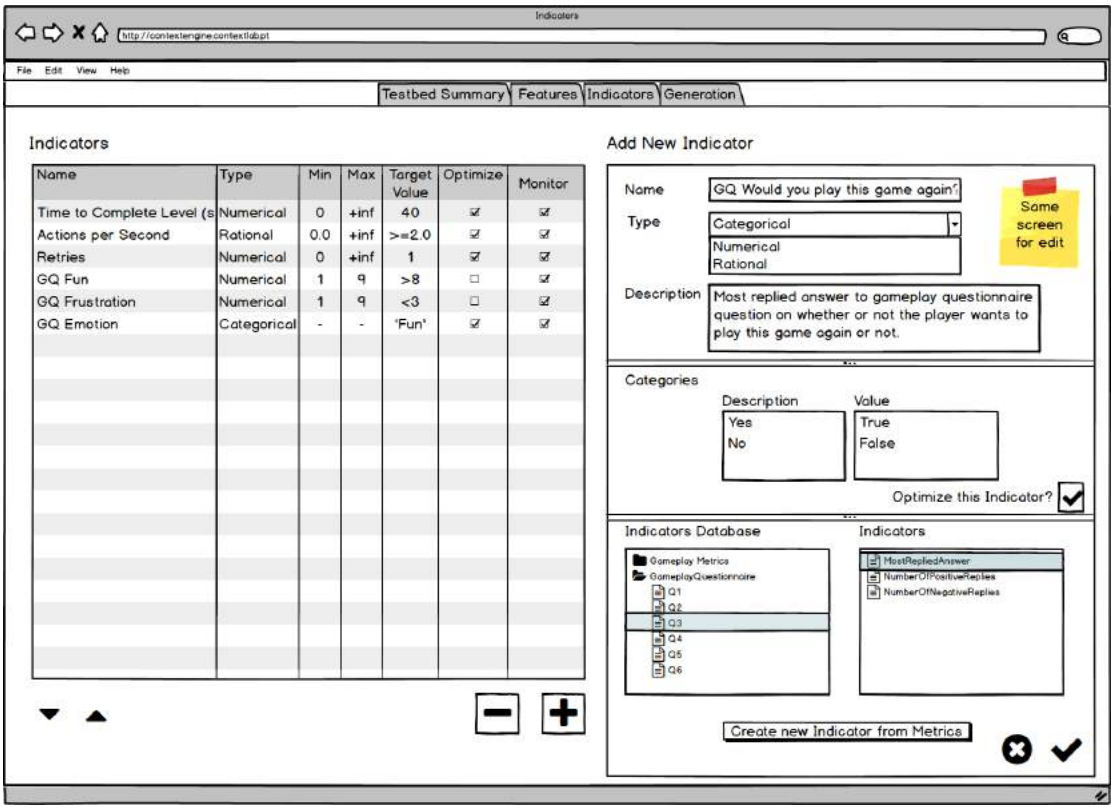

FIGURE A.7: This screen remained largely identical to its predecessor. Two minor additions: a description input box, and greater detail to the List pane data. The Metrics term was consistently replaced with Indicators, and references to the types of indicators (external vs internal, meaning subject reports and game-play metrics, respectively) were removed.

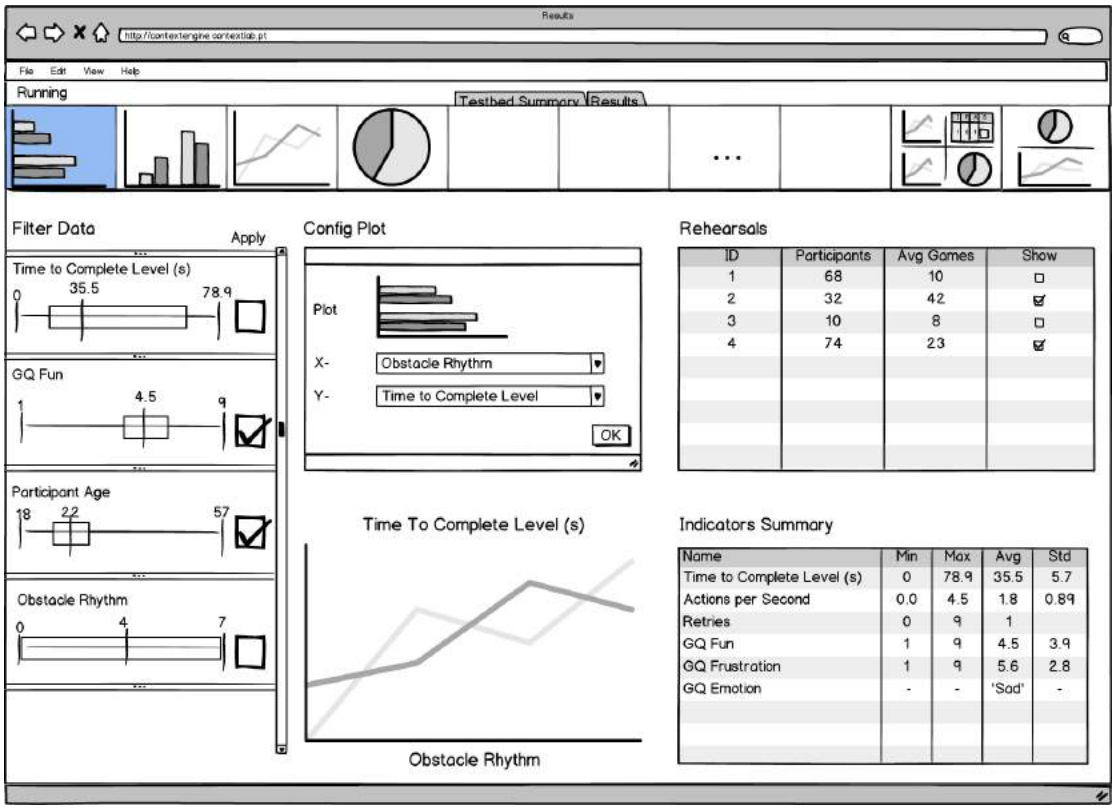

FIGURE A.8: The results screen was cleanly transcribed into digital form, maintaining its intricate layout and abundance of user-options.

A.3 Participatory Design 2

In the second Participatory Design session, designers took Prototype 1 and made several edits. They did not create new screens, inscribing their changes directly into the previous prototype (see figure A.9), or mapping them with numbers to new paper drawings (see figure A.10). Because it is more difficult to inspect the end result of these screens (that are mostly verbally annotated scribbles pointing to specific elements in the original interfaces), we only present one example. Of note here is the proposal for two new pages: one for deployment setup and another for user login and project assignment.

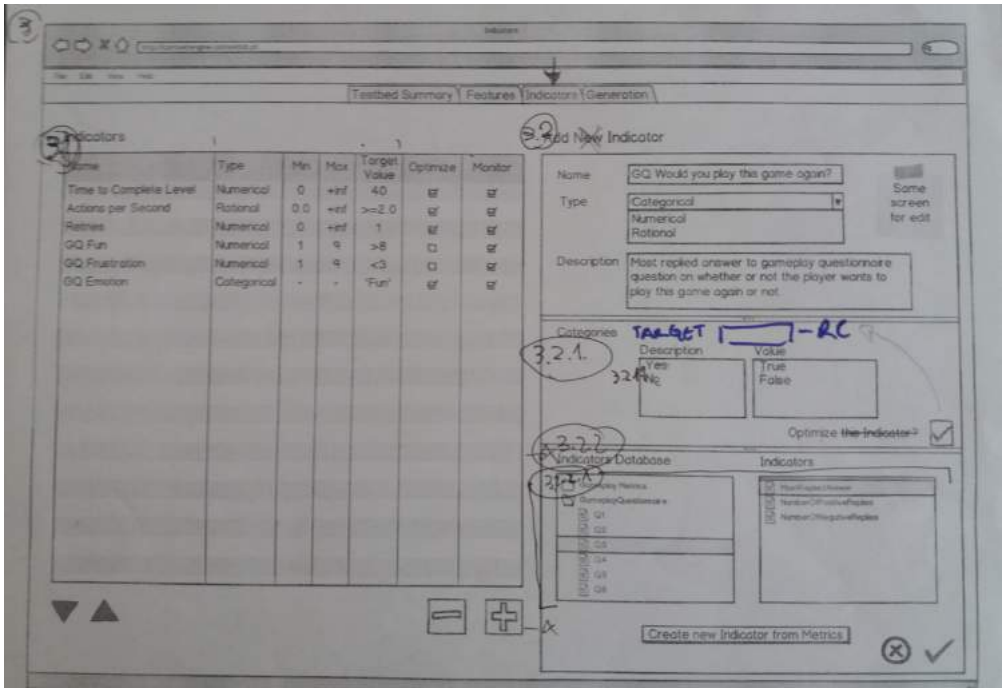

FIGURE A.9: Example of how Prototype 1's screens were annotated. In the next figure are the changes as proposed by designers.

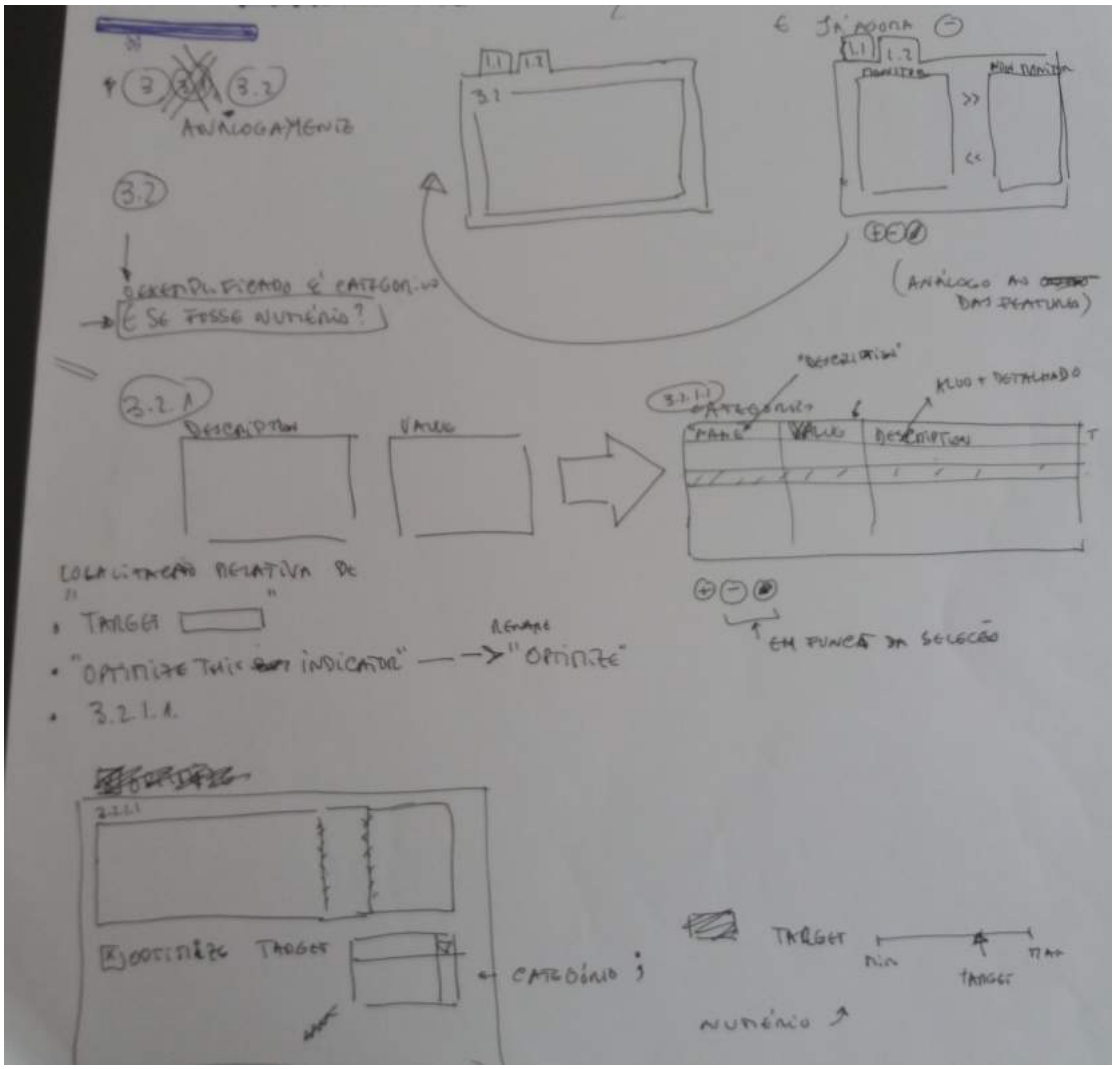

FIGURE A.10: Drawings made by Designers in Participatory Design session 2 as edits to the original. As an example, in the top right, you can see the change from the original version that had an overview page and two pages for editing of Features and Indicators, to a single Overview Page with 2 dynamic list panes. If the user wishes to edit Features the Indicator pane disappears and is replaced with the Original Indicator Edit pane, and vice versa.

A.4 Prototype 2

As mentioned in the main body of the thesis, the second version had to cope with several problems that were detected during the previous Participatory Design session. To solve these we changed the nomenclature of several interface items and introduced the Design Problem metaphor, with Game Variations and Design Goal Tests. Proposals from the Participatory design sessions were carried to the new version whenever possible.

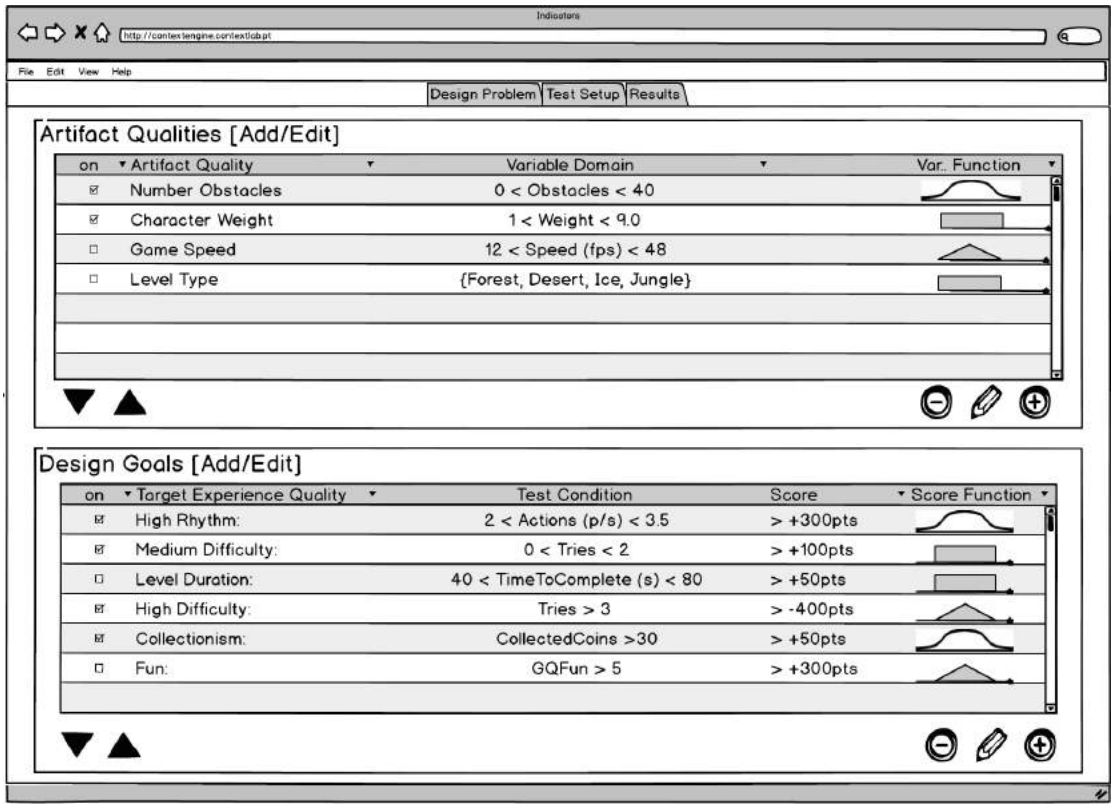

FIGURE A.11: The biggest change here from the previous version was the change from a two vertical panes to two horizontal panes. Tests with webpage layouts had revealed that these panes would be too cramped for content if aligned vertically, so there was a change to a horizontal orientation for panes. This makes the page vertically scrollable, which is more in line with typical web-page layouts. Despite this, their functionality remained as previously specified by designers, including their dynamic swapping.

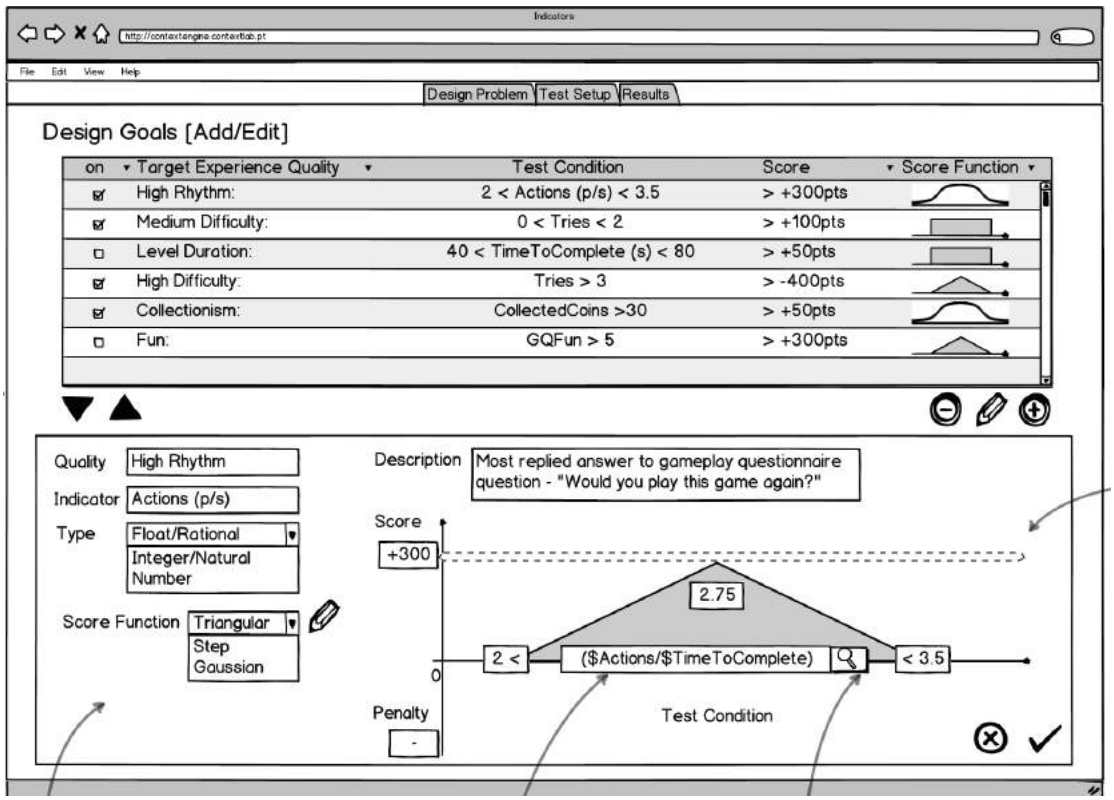

FIGURE A.12: This is where the major change in terms of interface and functioning happened, by way of the Design Goal test metaphor. All additions to the design were in trying to maintain the logic of the pre-existing layout.

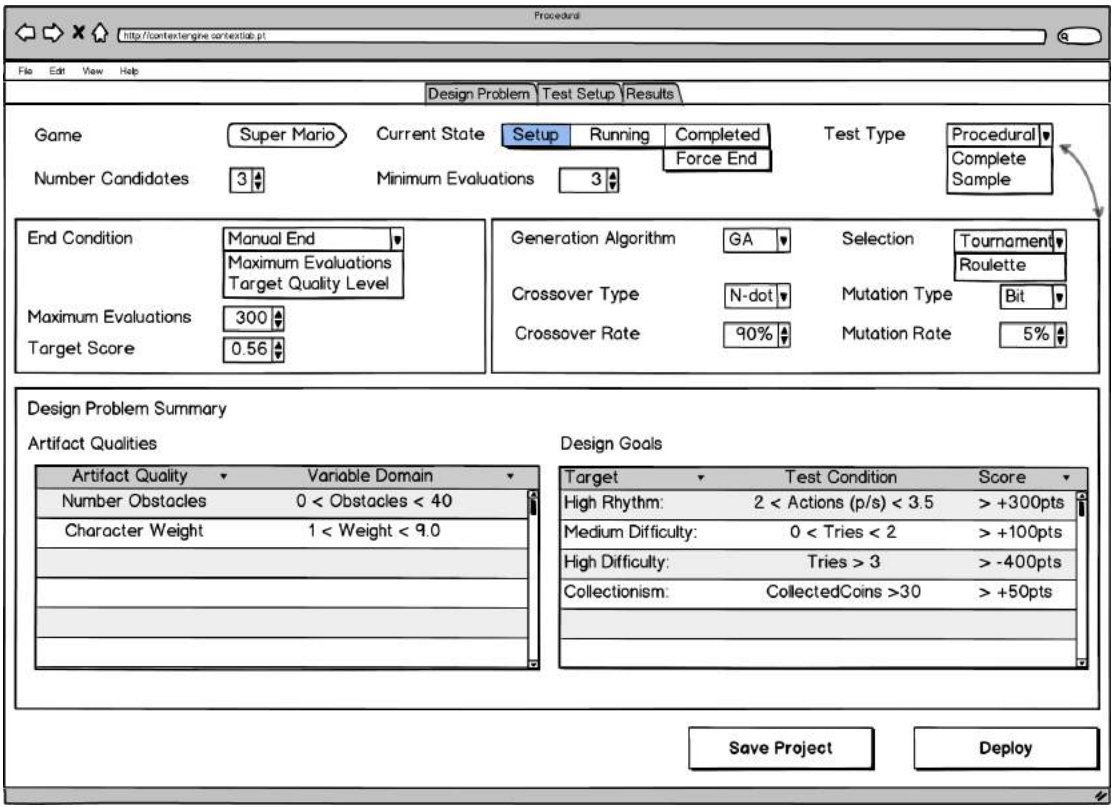

FIGURE A.13: This is the result of the PD2 proposal for a Setup page. During that session, parts of this screen were drawn, though the general layout was missing. We adapted the outline of previous pages and inserted the elements drawn in PD2.

A.5 Development Prototype 1

Prototype 2 was committed to development, and resulted in the prototype below. Note some features are not yet included for lack of development (specifically, the filtering engine originally proposed in Participatory Design 1).

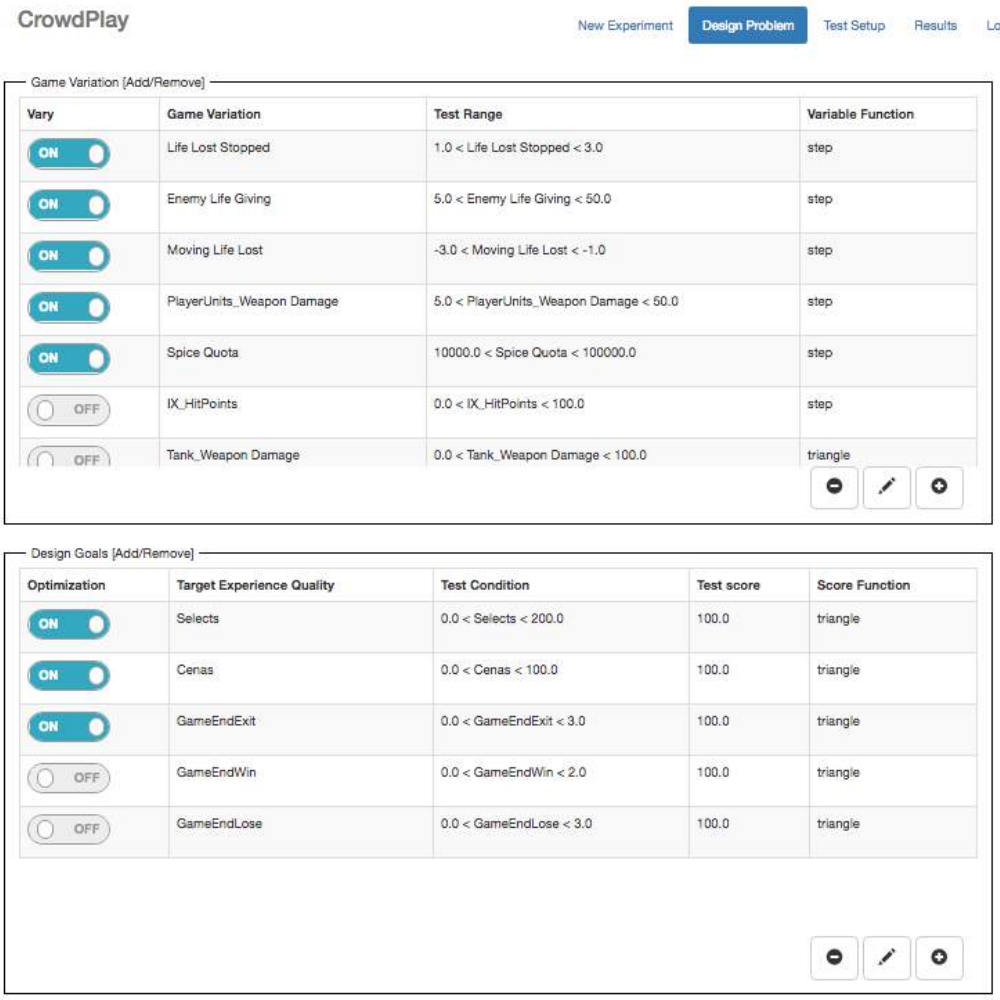

FIGURE A.14: Design Problem page. Its working is exactly as designed in PD2, with the exception of using two horizontal panes, instead of vertical ones.

FIGURE A.15: Adding a Game Variation or clicking in a Game Variation list element opens up the editing menu for that Variation.

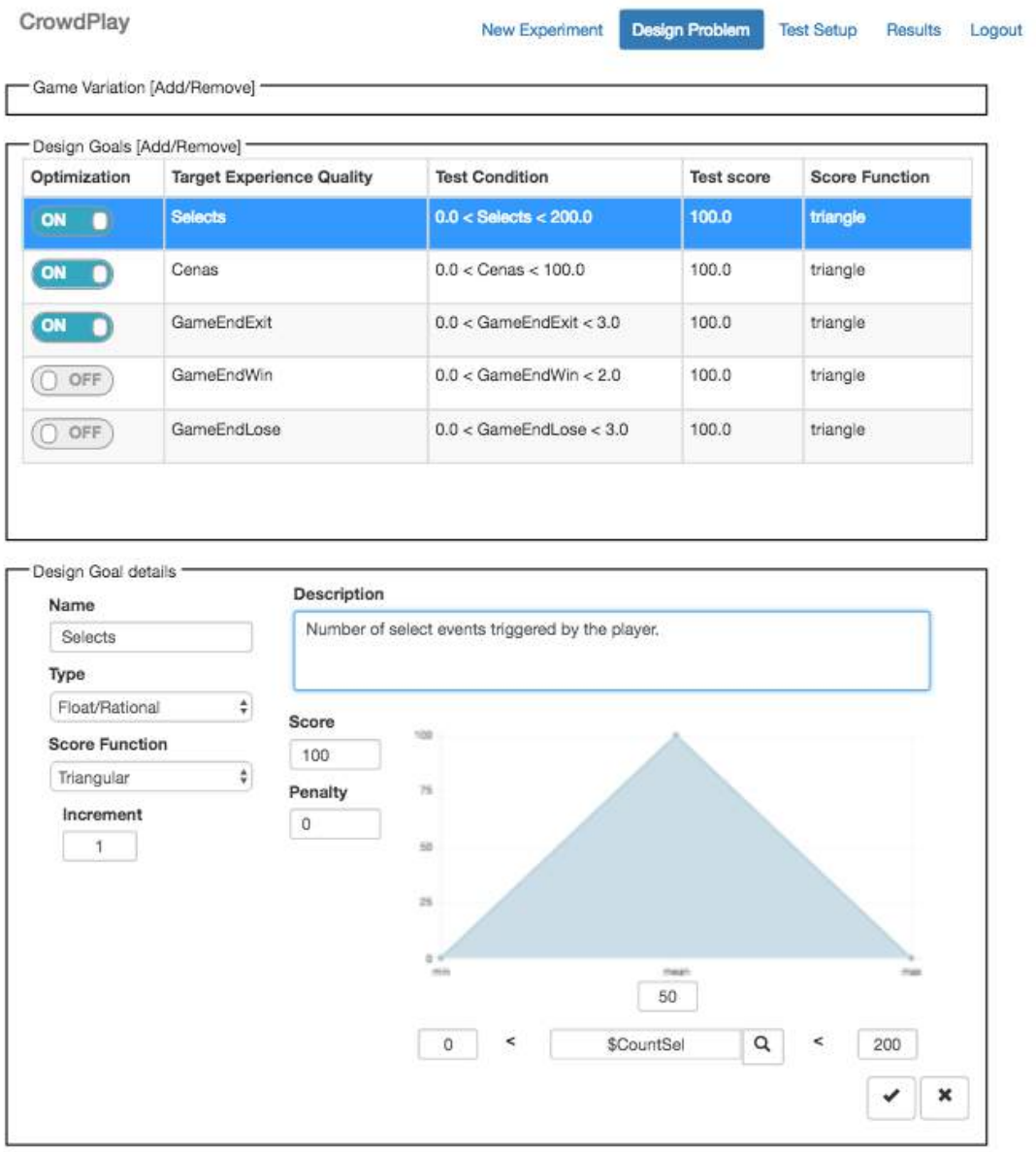

FIGURE A.16: Design Goals work analogously to Variations.

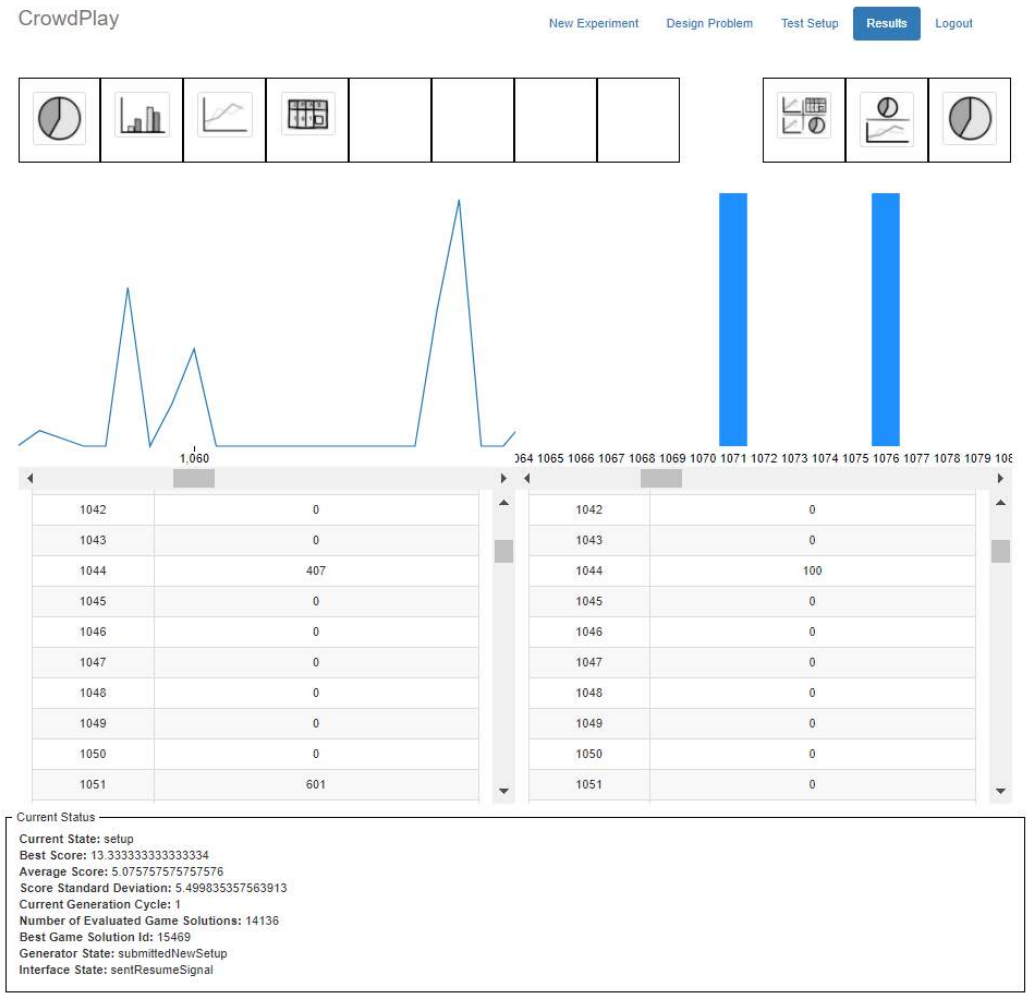

FIGURE A.17: The Results Page. Designers explicitly asked for a system for filtering data in function of all variables (both Game Parameters and Experience Indicators), but for lack of development time, in this prototype this feature was not yet implemented.

Bibliography

Adams, Tarn (2006). *Dwarf Fortress*. Bay 12 Games, Windows.

Alena Denisova A Imran Nordin, Paul Cairns (2016). In: *Conference Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play*, pp. 33–37.

Armstrong, Matthew (2009). *Borderlands*. Gearbox Software, Xbox 360.

Arseth, Espen (2001). "Computer Game Studies, Year One (editorial)". In: *the international journal of computer game research*. Ed. by Espen Arseth. Vol. 1.

Bartle, R.A. (2004). *Designing virtual worlds*. New Riders Games Series. New Riders.

Bertz, Matt (2011). *The Technology behind Elder Scrolls V: Skyrim*. Online; accessed February 1st 2016. URL: `http://www.gameinformer.com/games/the_elder_scrolls_v_skyrim/b/xbox360/archive/2011/01/17/the-technology-behind-elder-scrolls-v-skyrim.aspx`.

Bogost, Ian (2007). *Persuasive Games: The Expressive Power of Videogames*. Cambridge, Massachussets: The MIT Press.

Booth, Mike (2008). *Left 4 Dead*. Turtle Rock Studios, Xbox 360.

— (2009). *Left 4 Dead 2*. Turtle Rock Studios, Xbox 360.

Brabben, David and Ian Bell (1984). *Elite*. Acornsoft Limited, BBC Micro.

Brathwaite, B. and I. Schreiber (2009). *Challenges for game designers*. Course Technology.

Brevik, David and Erich Schaefer (1996). *Diablo*. Blizzard North, PC CD-ROM.

Britannica, Encyclopaedia (2008). *Britannica Concise Encyclopedia*. Encyclopaedia Britannica. ISBN: 9781593394929.

Börner, Katy and Shashikant Penumarthy (2003). "Social diffusion patterns in three-dimensional virtual worlds". In: *Information Visualization* 2, pp. 182–198.

Caillois, Roger (2001). *Man, Play and Games*. University of Illinois Press.

Calleja, Gordon (2007). "Digital games as designed experience: Reframing the concept of immersion". PhD thesis.

Cardoso, Amilcar, Tony Veale, and Geraint Wiggins (2009). "Converging on the Divergent: The history (and future) of the International Joint Workshops in Computational Creativity". In: *AI Magazine* 30.3. Ed. by Oliviero Stock Ramon Lopez de Mantaras and Simon Colton.

Castle, Louis (1997). *Blade Runner*. Westwood Studios, PC CD-ROM.

Chanel, Guillaume et al. (2006). "Emotion assessment: arousal evaluation using EEG's and peripheral physiological signals". In: *Final Project Report Proc. Int. Workshop on Multimedia Content Representation, Classification and Security*. Dubrovnik, Croatia, pp. 530–537.

Chen, Jenova (2007). "Flow in Games". accessed August 2009. URL: http://www.jenovachen.com/flowingames/Flow_in_games_final.pdf.

Cherry, Erin and Celine Latulipe (2014). "Quantifying the Creativity Support of Digital Tools Through the Creativity Support Index". In: *ACM Trans. Comput.-Hum. Interact.* 21.4.

Chittaro, Luca, Roberto Ranon, and Lucio Ieronutti (2006). "VU-Flow: A Visualization Tool for Analyzing Navigation in Virtual Environments". In: *IEEE Transactions on Visualization and Computer Graphics* 12, pp. 1475–1485. ISSN: 1077-2626. DOI: http://doi.ieeecomputersociety.org/10.1109/TVCG.2006.109.

Colton, Simon and Geraint A. Wiggins (2012). "Computational Creativity: The Final Frontier?" In: *ECAI*. Ed. by Luc De Raedt et al. Vol. 242. Frontiers in Artificial Intelligence and Applications. IOS Press, pp. 21–26.

Cook, Michael and Simon Colton (2014). "Ludus Ex Machina: Building A 3D Game Designer That Competes Alongside Humans". In: *Proceedings of the Fifth International Conference on Computational Creativity*. ICCC2014. International Association for Computational Creativity.

Costello, Brigid and Ernest Edmonds (2009). "A Tool for Characterizing the Experience of Play". In: *Proceedings of the Sixth Australasian Conference on Interactive Entertainment*. IE '09. ACM.

Costikyan, Greg (2005). *Death to the Games Industry*. Escapist Magazine. accessed July, 2011. URL: http://www.escapistmagazine.com/articles/view/issues/issue_8/50-Death-to-the-Games-Industry-Part-I.6.

Crabtree, Andy (1998). "Ethnography in Participatory Design". In: *Proceedings of the 1998 Participatory Design Conference*. Computer Professionals for Social Responsibility, pp. 93–105.

Craveirinha, Rui, and Licinio Roque (2016). "Exploring the Design-Space: The Authorial Game Evolution Tool Case-Study". In: *Proceedings of the 2016 Advances in Computer Entertainment Conference*. Osaka, Japan: ACM.

Craveirinha, Rui, Nuno Barreto, and Licinio Roque (2016). "Towards a Taxonomy for the Clarification of PCG Actors' Roles". In: *Proceedings of the 2016 Annual*

Symposium on Computer-Human Interaction in Play. CHI PLAY '16. Austin, Texas, USA: ACM, pp. 244–253.

Craveirinha, Rui and Licinio Roque (2010). "Looking for the Heart of Interactive Media - Reflections on Video Games' Emotional Expression". In: *Fun and Games*. Leuven, Belgium.

— (2011). "Zero Lecture in Game Design". In: *Proceedings of SBGames 2011 Arts and Design Track - Full Papers*. Salvador, Brazil.

— (2015a). "Designing Games with Procedural Content Generation: An Authorial Approach". In: *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*. CHI EA '15. Seoul, Republic of Korea: ACM, pp. 1199–1204.

— (2015b). "Studying an Author-Oriented Approach to Procedural Content Generation through Participatory Design". In: *Entertainment Computing - ICEC 2015, 14th International Conference Proceedings*, pp. 383–390.

Craveirinha, Rui, Lucas Santos, and Licínio Roque (2013). "An Author-Centric Approach to Procedural Content Generation". In: *Advances in Computer Entertainment: 10th International Conference, ACE 2013, Boekelo, The Netherlands, November 12-15, 2013. Proceedings*. Ed. by Dennis Reidsma, Haruhiro Katayose, and Anton Nijholt. Springer International Publishing, pp. 14–28.

Crawford, Chris (1982). *The Art of Computer Game Design*. Vancouver: Washington State University.

Creswell, John W. (2013). *Research Design: Qualitative, Quantitative, & Mixed Methods Approaches*. 4th ed. SAGE Publications, Inc.

Crilly, Nathan, Anja Maier, and P. John Clarkson (2008). "Representing Artefacts as Media: Modelling the Relationship Between Designer Intent and Consumer Experience". In: *International Journal of Design* 2.3, pp. 15–27.

Cross, Nigel (1999). "Design Research: A Disciplined Conversation". In: *Design Issues* 15.2, pp. 5–10.

Csíkszentmihályi, Mihály (1990). *Flow: the Psychology of Optimal Experience*. New York: Harper Perennial.

— (1997). *Creativity: flow and the psychology of discovery and invention*. Harper-Perennial.

Derakhshan, A., F. Hammer, and H.H. Lund (2006). "Adapting Playgrounds for Children's Play using Ambient Playware". In: *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pp. 5625–5630.

Dillon, Cath et al. (2000). "Aroused and Immersed: The Psychophysiology of Presence". MA thesis. UK: Goldsmith College / Independent Television Commission.

Drachen, Anders and Alessandro Canossa (2009). "Towards gameplay analysis via gameplay metrics". In: *Proceedings of the 13th International MindTrek Conference: Everyday Life in the Ubiquitous Era*. MindTrek '09. ACM, pp. 202–209.

Drachen, Anders, Alessandro Canossa, and Georgios N. Yannakakis (2009). "Player modeling using self-organization in Tomb Raider: Underworld". In: *CIG'09: Proceedings of the 5th international conference on Computational Intelligence and Games*. IEEE Press, pp. 1–8.

Ekman, Paul (1999). "Basic Emotion". In: *Handbook of Cognition and Emotion*. Ed. by Tim Dalgleish and Mick J. Power. John Wiley & Sons.

Ermi, Laura and Frans Mayra (2005). "Fundamental Components of the Gameplay Experience: Analysing Immersion". In: *Proceedings of the 2005 DiGRA International Conference: Changing Views: Worlds in Play*.

Faulkner, Laura (2003). "Beyond the five-user assumption: Benefits of increased sample sizes in usability testing". In: *Behavior Research Methods, Instruments, & Computers* 35.3.

Figueiredo, António Dias and Paulo Rupino Cunha (2007). "Information Systems Action Research: An Applied View of Emerging Concepts and Methods". In: ed. by Ned Kock. Springer US. Chap. Action Research and Design in Information Systems.

Frasca, Gonzalo (2001). "Videogames of the Oppressed - Videogames as a means for critical thinking and debate". MA thesis. School of Literature, Communication and Culture of the Georgia Institute of Technology.

Frayling, C. (1993). "Research in Art and Design". In: *Royal College of Art Research Papers* 1.1, pp. 1–5.

Frijda, Nico H. (2008). "The Psychologists' Point of View". In: *Handbook of Emotions, Third Edition*. Ed. by Michael Lewis, Jeannette M. Haviland-Jones, and Lisa Feldman Barrett. The Guilford Press.

Fullerton, T., C. Swain, and S. Hoffman (2008). *Game design workshop: a playcentric approach to creating innovative games*. Gama Network Series. Elsevier Morgan Kaufmann.

Gladwell, M. (2005). *Blink: The Power Of Thinking Without Thinking*. Little, Brown and Company.

Gombrich, E.H. (2009). *The story of art*. Phaidon.

Gregor, Shirley and Alan R. Hevner (June 2013). "Positioning and Presenting Design Science Research for Maximum Impact". In: *MIS Q.* 37.2, pp. 337–356. ISSN: 0276-7783.

Gregor, Shirley and David Jones (2007). "The anatomy of a design theory". In: *Journal of the Association of Information Systems*, pp. 312–335.

Gross, James J. (1999). "Emotion and Emotion Regulation". In: *Handbook of Personality: Theory and Research, 2nd Edition*. Ed. by Oliver P. John and Lawrence A. Pervin. New York / London: Guilford Press, pp. 525 –552.

Gross, James J. and Robert W. Levenson (1995). "Emotion Elicitation Using Films". In: *Cognition and Emotion* 9.1, pp. 87 –108.

Haag, Andreas et al. (2004). "Emotion Recognition Using Bio-Sensors: First Steps Towards an Automatic System". In: *Lecture Notes in Computer Science: Affective Dialogue Systems*. Vol. 3068/2004. Springer Berlin / Heidelberg, pp. 36 –48.

Harrison, Grant and Jason Kingsley (1990). *Murder!* U.S. GOLD, Amiga C64.

Hassenzahl, Marc (2011). "User Experience and Experience Design". In: *Encyclopedia of Human-Computer Interaction*. Ed. by Mads Soegaard and Rikke Friis Dam. Interaction-Design.org.

Hastings, Erin J. and Kenneth O. Stanley (2010). "Galactic Arms Race: An Experiment in Evolving Video Game Content". In: *SIGEVOlution* 4.4.

Haupt, Randy L. and Sue Ellen Haupt (2004). *Practical genetic algorithms*. Second. John Wiley & Sons, Inc.

Healey, Mark and David Smith (2008). *Little Big Planet*. Media Molecule, Playstation 3.

Heinbokel, T et al. (1996). "Don't underestimate the problems of user centredness in software development projects - There are many!" In: *Behaviour Information Technology* 15.4, pp. 226–236.

Hendrikx, Mark et al. (2013). "Procedural Content Generation for Games: A Survey". In: *ACM Trans. Multimedia Comput. Commun. Appl.* 9.1, 1:1–1:22.

Herman, Leonard (2001). *Phoenix: The Fall & Rise of Videogames*. 3rd ed. ROLENTA PRESS.

Hevner, Alan R. et al. (Mar. 2004). "Design Science in Information Systems Research". In: *MIS Q.* 28.1, pp. 75–105.

Hoobler, Nate, Greg Humphreys, and Maneesh Agrawala (2004). "Visualizing Competitive Behaviors in Multi-User Virtual Environments". In: *In Proc. Viz'04*, pp. 163–170.

Horlings, Robert (2008). "Emotion recognition using brain activity". MA thesis. Man-machine interaction group, Delft University of Technology.

Howard, Todd (2011). *The Elder Scrolls V: Skyrim*. Bethesda Game Studios, Xbox 360.

— (2015). *Fallout 4*. Bethesda Game Studios, Playstation 4.

Howell, Ryan T. and Graham Hill (2009). "The mediators of experiential purchases: Determining the impact of psychological needs satisfaction and social comparison". In: *The Journal of Positive Psychology* 4.6, pp. 511–522.

Hunicke, Robin, Marc Leblanc, and Robert Zubek (2004). "MDA: A formal approach to game design and game research". In: *In Proceedings of the Challenges in Games AI Workshop, Nineteenth National Conference of Artificial Intelligence*. Press.

Huta, Veronika and Richard Ryan (2010). "Pursuing Pleasure or Virtue: The Differential and Overlapping Well-Being Benefits of Hedonic and Eudaimonic Motives". In: *Journal of Happiness Studies* 11.6, pp. 735–762.

IDV (2016). *SpeedTree*. Online; accessed February 1st 2016. URL: `http://www.speedtree.com/`.

Irish, Dan (2005). *The Game Producer's Handbook*. Boston, MA, United States: Course Technology Press.

Isbister, Katherine et al. (2006). "The Sensual Evaluation Instrument: Developing an Affective Evaluation Tool". In: *In Proc. CHI '06*. ACM Press, pp. 1163–1172.

Jenkins, Sean, Raymond Brown, and Neil Rutterford (2009). "Comparing Thermographic, EEG, and Subjective Measures of Affective Experience During Simulated Product Interactions". In: *International Journal of Design* 3.2.

Johnson, Jeff and Austin Henderson (2002). "Conceptual Models: Begin by Designing What to Design". In: *interactions* 9.1, pp. 25–32.

Juul, Jesper (1999). "A Clash between Game and Narrative". MA thesis. Cambridge, Massachussets: Institute of Nordic Language and Literature, University of Copenhagen.

— (2005). *Half Real: Video Games between Real Rules and Fictional Worlds*. Cambridge, Massachussets: MIT Press.

Katz, Melissa R. (1995). "William Holman Hunt and the 'Pre-Raphaelite Technique'". In: *Historical Painting Techniques, Materials, and Studio Practice: Preprints of a Symposium Held at the University of Leiden, the Netherlands 26-29 June, 1995*. Getty Conservation Institute.

Keith, Clinton (2010). *Agile Game Development with Scrum*. 1st. Addison-Wesley Professional.

Kensing, Finn and Jeanette Blomberg (1998). "Participatory Design: Issues and Concerns". In: *Comput. Supported Coop. Work* 7.3-4, pp. 167–185. ISSN: 0925-9724.

Kensing, Finn, Jesper Simonsen, and Keld Bodker (1998). "MUST: A Method for Participatory Design". In: *Human-Computer Interaction* 13.2, pp. 167–198.

Kerssemakers, M. et al. (2012). "A procedural procedural level generator generator". In: *Computational Intelligence and Games (CIG), 2012 IEEE Conference on*.

Khalili, Z. and M.H. Moradi (2008). "Emotion detection using brain and peripheral signals". In: *Biomedical Engineering Conference, 2008. CIBEC 2008. Cairo International*, pp. 1–4.

Kim, Jun H. et al. (2008). "Tracking real-time user experience (TRUE): a comprehensive instrumentation solution for complex systems". In: *Computer Human Interaction*, pp. 443–452.

Ko, Kwang-Eun, Hyun-Chang Yang, and Kwee-Bo Sim (2009). "Emotion recognition using EEG signals with relative power values and Bayesian network". In: *International Journal of Control, Automation and Systems* 7.5, pp. 865–870.

Koster, R. (2005). *A theory of fun for game design*. Paraglyph Series. Paraglyph Press.

Krippendorff, Klaus (1989). "On the Essential Contexts of Artifacts or on the Proposition That Design Is Making Sense (Of Things)". In: *Design Issues* 5.2.

Kujala, Sari (2003). "User involvement: A review of the benefits and challenges". In: *Behaviour & Information Technology* 22.1, pp. 1–16.

Lankveld, Giel van et al. (2009). "Incongruity-Based Adaptive Game Balancing." In: *ACG'09*, pp. 208–220.

Lazar, Jonathan, Jinjuan Feng, and Harry Hochheiser (2010). *Research Methods in Human-Computer Interaction*. Wiley. ISBN: 978-0-470-72337-1.

Lazzaro, Nicole (2004). "Why We Play Games: Four Keys to More Emotion Without Story". In: *Game Developers Conference*. URL: `http://xeodesign.com/xeodesign_whyweplaygames.pdf`.

Lee, Teddy (2015). *Rogue Legacy*. Cellar Door Games, PS4 digital.

Li, Mu and Bao-Liang Lu (2009). "Emotion Classification Based on Gamma-band EEG". In: *Poster Session Biomedical Signals and Systems II, 31st Annual International IEEE EMBS Conference*. Minnesota, USA.

Liapis, Antonios, Georgios N. Yannakakis, and Julian Togelius (2013). "Sentient Sketchbook: Computer-Aided Game Level Authoring". In: *Proceedings of the 8th Conference on the Foundations of Digital Games*.

Liapis, Antonios, Georgios N. Yannakakis, and Julian Togelius (2014). "Designer Modeling for Sentient Sketchbook". In: *Proceedings of the IEEE Conference on Computational Intelligence and Games (CIG)*.

Löwgren, Jonas (1995). "Applying Design Methodology to Software Development". In: *Proceedings of the 1st Conference on Designing Interactive Systems: Processes, Practices, Methods, & Techniques*. DIS '95. ACM, pp. 87–95.

LSDWiki (2016). *LSD: Dream Emulator*. Online; accessed February 1st 2016. URL: `http://dreamemulator.wikia.com/`.

Lubart, Todd (2005). "How can computers be partners in the creative process: Classification and commentary on the Special Issue". In: *International Journal of Human-Computer Studies* 63.4?5, pp. 365 –369.

Machado, Penousal and Hugo Amaro (2013). "Fitness Functions for Ant Colony Paintings". In: *Proceedings of the fourth International Conference on Computational Creativity (ICCC)*.

Machado, Penousal et al. (2004). "Applications of Evolutionary Computing EvoWorkshops Proceedings". In: ed. by Günther R. Raidl et al. Springer Berlin Heidelberg. Chap. Adaptive Critics for Evolutionary Artists.

Machado, Penousal et al. (2014). "Evolutionary and Biologically Inspired Music, Sound, Art and Design: Third European Conference, EvoMUSART". In: ed. by Juan Romero, James McDermott, and João Correia. Springer Berlin Heidelberg. Chap. An Interface for Fitness Function Design.

Mahlmann, Tobias et al. (2010). "Predicting player behavior in Tomb Raider: Underworld". In: *CIG*, pp. 178–185.

Martin, Andrew et al. (2010). "Applications of Evolutionary Computation EvoApplicatons, Proceedings, Part I". In: ed. by Cecilia Chio et al. Springer Berlin Heidelberg. Chap. Evolving 3D Buildings for the Prototype Video Game Subversion.

Mateas, Michael and Andrew Stern (2003). "Facade: An Experiment in Building a Fully-Realized Interactive Drama". In: *Game Developers Conference*.

McCarthy, J. and P. Wright (2004). *Technology as experience*. MIT Press.

McMillen, Edmund (2014). *Binding of Isaac: Rebirth*. Nicalis, PS4 digital.

Meier, Sid (2005). *Civilization IV*. Firaxis Games, PC.

Moura, Arnaldo (2011). "Recycle Report". MA thesis. Coimbra, Portugal: Science and Technology Faculty of the University of Coimbra.

Muller, Michael J. (1991). "PICTIVE&Mdash;an Exploration in Participatory Design". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '91. ACM, pp. 225–231.

Murray, Janet H. (2011). *Inventing the Medium: Principles of Interaction Design As a Cultural Practice*. The MIT Press.

Nacke, Lennart E. et al. (2009). "Playability and Player Experience Research [Panel Abstracts]". In: *Breaking New Ground: Innovation in Games, Play, Practice and Theory: Proceedings of the 2009 Digital Games Research Association Conference*. Ed. by Atkins Barry, Kennedy Helen, and Krzywinska Tanya. Brunel University.

Nelson, Mark J. and Michael Mateas (2007). "Towards Automated Game Design". In: *Congress of the Italian Association for Artificial Intelligence*.

Pedersen, C., J. Togelius, and G.N. Yannakakis (2010). "Modeling Player Experience for Content Creation". In: *Computational Intelligence and AI in Games, IEEE Transactions on* 2.1, pp. 54 –67.

Pedersen, Chris, Julian Togelius, and Georgios N. Yannakakis (2009). "Modeling player experience in super mario bros". In: *Proceedings of the 5th international conference on Computational Intelligence and Games*. CIG'09. IEEE Press, pp. 132–139.

Pereira, L.L. and Licinio Roque (2012). "Towards a game experience design model centered on participation". In: *The 30th ACM Conference on Human Factors in Computing Systems CHI2012 Extended Abstracts*.

Pereira, Luís Lucas and Licinio Roque (2013a). "Gameplay Experience Evaluation Centered on Participation: The FáTima Game Design Case". In: *CHI '13 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '13. ACM.

Pereira, Luís Lucas and Licinio Roque (2013b). "Human Factors in Computing and Informatics: First International Conference, SouthCHI 2013, Maribor, Slovenia, July 1-3, 2013. Proceedings". In: ed. by Andreas Holzinger et al. Springer Berlin Heidelberg. Chap. A Preliminary Evaluation of a Participation-Centered Gameplay Experience Design Model.

Persson, Markus (2011). *Minecraft*. Mojang, PC Download.

Piselli, Paolo, Mark Claypool, and James Doyle (2009). "Relating cognitive models of computer games to user evaluations of entertainment". In: *Foundations of Digital Games*, pp. 153–160. DOI: 10.1145/1536513.1536545.

Plutchik, Robert (2001). "The nature of emotions". In: *American Scientist* 89, pp. 344 –350.

Poole, Steven (2000). *Trigger Happy*. New York: Arcade Publishing.

QuadSoftware (2016). *SpeedTree*. http://www.quadsoftware.com/. Online; accessed February 1st 2016.

Rigby, Scott and Richard Ryan (2007). *The Player Experience of Need Satisfaction(PENS): anÊappliedÊmodelÊandÊmethodologyÊforÊunderstandingÊkey componentsÊofÊtheÊplayerÊexperience*. Tech. rep. Celebration, FL: Immersyve.

Rottenberg, Jonathan, Rebecca D. Ray, and James J. Gross (2007). "Emotion Elicitation Using Films". In: *Handbook of emotion elicitation and assessment*. US: Oxford University Press, pp. 9 –28.

Ryan, R. M. and E. L. Deci (2001). "On happiness and human potentials: a review of research on hedonic and eudaimonic well-being". In: *Annual Review of Psychology* 52, pp. 141–166.

Salen, Katie and Eric Zimmerman (2004). *Rules of Play: Game Design Fundamentals*. Cambridge, Massachussets: The MIT Press.

Sato, Osamu (1998). *LSD*. Asmik Ace Entertainment, Playstation.

Schell, Jesse (2008). *The Art of Game Design: A book of lenses*. Morgan Kaufmann.

Schuler, Douglas and Aki Namioka, eds. (1993). *Participatory Design: Principles and Practices*. L. Erlbaum Associates Inc.

Sears, A. and J.A. Jacko (2009). *Human-Computer Interaction Development Process*. Human Factors and Ergonomics. Taylor & Francis. ISBN: 9781420088892.

Shaker, Noor, Georgios N. Yannakakis, and Julian Togelius (2010). "Towards Automatic Personalized Content Generation for Platform Games". In: *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*. AAAI Press.

Shneiderman, Ben (Dec. 2007). "Creativity Support Tools: Accelerating Discovery and Innovation". In: *Commun. ACM* 50.12, pp. 20–32. ISSN: 0001-0782.

Shneiderman, Ben et al. (2006). "Creativity Support Tools: Report From a U.S. National Science Foundation Sponsored Workshop". In: *International Journal of Human-Computer Interaction* 20.2.

Smith, Gillian (2012). "Expressive Design Tools: Procedural Content Generation for Game Designers". PhD thesis. UC Santa Cruz.

— (2014). "Expressive Design Tools: Procedural Content Generation for Game Designers". In: *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*.

Smith, Gillian, Jim Whitehead, and Michael Mateas (2010a). "Tanagra: a mixed-initiative level design tool". In: *Proceedings of the Fifth International Conference on the Foundations of Digital Games*. FDG '10. ACM, pp. 209–216.

— (2010b). "Tanagra: An Intelligent Level Design Assistant for 2D Platformers." In: *AIIDE*. The AAAI Press.

— (2011). "Tanagra: Reactive Planning and Constraint Solving for Mixed-Initiative Level Design." In: *IEEE Trans. Comput. Intellig. and AI in Games* 3.3, pp. 201–215.

Smith, Gillian et al. (2012). "PCG-based Game Design: Creating Endless Web". In: *Proceedings of the International Conference on the Foundations of Digital Games*. FDG '12. ACM.

Smith, Randy (May 2008). *The Tyranny of Fun and of Lord Blackthorn*.

Software, Planetside (2016). *Terragen 3*. Online; accessed February 1st 2016. URL: `planetside.co.uk/`.

Spufford, Francis (2003). *Masters of their universe*. Online; accessed February 1st 2016. URL: `hhttp://www.theguardian.com/books/2003/oct/18/features.weekend`.

Steen, Marc, Lottie Kuijt-Evers, and Jente Klok (2007). "Early user involvement in research and design projects – A review of methods and practices". In: *23rd EGOS Colloquium (European Group for Organizational Studies)*. Chap. Theme 25: Dancing with users: How to organize innovation with consumers and users?

Sweetser, Penelope and Peta Wyeth (2005). "GameFlow: a model for evaluating player enjoyment in games". In: *Comput. Entertain.* 3 (3), pp. 3–3.

Takagi, Hideyuki (2001). "Interactive Evolutionary Computation: Fusion of the Capabilities of EC Optimization and Human Evaluation". In: *Proceedings of the IEEE 89.9*.

Takahashi, Kazuhiko (2004). "Remarks on Emotion Recognition from Bio-Potential Signals". In: *2nd International Conference on Autonomous Robots and Agents*. New Zealand.

Tavinor, G. (2009). *The art of videogames*. New directions in aesthetics. Wiley-Blackwell. ISBN: 9781405187893.

Togelius, J. and J. Schmidhuber (2009). "An experiment in automatic game design". In: *IEEE Symposium On Computational Intelligence and Games (CIG) 2008*. IEEE, pp. 111–118.

Togelius, Julian, Renzo De Nardi, and Simon M. Lucas (2006). "Making racing fun through player modeling and track evolution". In: *in Proceedings of the SAB'06 Workshop on Adaptive Approaches for Optimizing Player Satisfaction in Computer and Physical Games*, p. 70.

Togelius, Julian et al. (2011). "Search-Based Procedural Content Generation: A Taxonomy and Survey". In: *Computational Intelligence and AI in Games, IEEE Transactions on* 3.3.

Tomaszkiewicz, Konrad, Mateusz Kanik, and Sebastian Stepien (2015). *Witcher 3: Wild Hunt*. CD Projekt Red, PS4.

Toy, Michael and Glenn Wichman (1984). *Rogue*. Artificial Intelligence Design, DOS.

Tullis, Tom and Bill Albert (2008). *Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics (Interactive Technologies): Collecting, Analyzing, and Presenting ... Kaufmann Series in Interactive Technologies)*. Morgan Kaufmann.

Tutenel, Tim et al. (2009). "Rule-based layout solving and its application to procedural interior generation". In: *Proceedings of the CASA Workshop on 3D Advanced Media In Gaming And Simulation (3AMIGAS)*.

Tychsen, Anders and Alessandro Canossa (2008). "Defining personas in games using metrics". In: *Proceedings of the 2008 Conference on Future Play: Research, Play, Share*. Future Play '08. ACM, pp. 73–80.

Vaishnavi, V. and W. Kuechler (2004). URL: http://www.desrist.org/design-research-in-information-systems/.

Van Der Panne, Gerben and Cees Beers (2003). "Success and Failure of Innovation: A Literature Review". In: *International Journal of Innovation Management* 7.3, pp. 309–338.

Vankleef, E, H Vantrijp, and Pieternel Luning (2005). "Consumer research in the early stages of new product development: a critical review of methods and techniques". In: *Food Quality and Preference* 16.3, pp. 181–201.

Vorderer, Peter, Tilo Hartmann, and Christoph Klimmt (2003). "Explaining the enjoyment of playing video games: the role of competition". In: *Proceedings of the Second International Conference on Entertainment Computing*.

Weber, Ben G. and Michael Mateas (2009). "A data mining approach to strategy prediction". In: *Proceedings of the 5th international conference on Computational Intelligence and Games*. CIG'09. IEEE Press, pp. 140–147.

WikiA (2016). *Left4Dead*. Online; accessed February 1st 2016. URL: http://left4dead.wikia.com/wiki/The_Director.

Wikipedia. *Left4Dead*. URL: https://en.wikipedia.org/wiki/Left_4_Dead.

— *Left4Dead2*. URL: https://en.wikipedia.org/wiki/Left_4_Dead_2.

— (2016a). *Minecraft*. Online; accessed February 1st 2016. URL: https://en.wikipedia.org/wiki/Minecraft.

— (2016b). *Terragen*. Online; accessed February 1st 2016. URL: https://en.wikipedia.org/wiki/Terragen.

— (2016c). *The Elder Scrolls V: Skyrim*. Online; accessed February 1st 2016. URL: https://en.wikipedia.org/wiki/The_Elder_Scrolls_V:_Skyrim.

Wilson, T D and J W Schooler (1991). "Thinking too much: introspection can reduce the quality of preferences and decisions." In: *Journal of Personality and Social Psychology* 60.2, pp. 181–192.

Wolf, Tracee Vetting et al. (2006). "Dispelling "Design" As the Black Art of CHI". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '06. ACM, pp. 521–530.

Yannakakis, G. N. (2008). "How to Model and Augment Player Satisfaction: A Review". In: *Proceedings of the 1st Workshop on Child , Computer and Interaction, ICMI'08,*

Yannakakis, G. N. and J. Hallam (2009). "Real-Time Game Adaptation for Optimizing Player Satisfaction". In: *IEEE Transactions on Computational Intelligence and Ai in Games* 1.

Yannakakis, Georgios and John Hallam (2005). "A Generic Approach for Generating Interesting Interactive Pac-Man Opponents". In: *In Proceedings of the IEEE Symposium on Computational Intelligence and Games*, pp. 94–101.

Yannakakis, Georgios N. and John Hallam (2004). "Evolving Opponents for Interesting Interactive Computer Games". In: *Simulation of Adaptive Behavior*.

— (2006). "Towards Capturing and Enhancing Entertainment in Computer Games". In: *SETN*.

— (2007). "Towards Optimizing Entertainment in Computer Games". In: *Appl. Artif. Intell.* 21 (10), pp. 933–971.

Yannakakis, Georgios N, Antonios Liapis, and Constantine Alexopoulos (2014). "Mixed-initiative co-creativity". In: *Proceedings of the ACM Conference on Foundations of Digital Games*.

Yannakakis, Georgios N. and Julian Togelius (2011). "Experience-Driven Procedural Content Generation". In: *IEEE Transactions on Affective Computing* 99.

Yu, Derek (2014). *Spelunky*. Mossmouth, PS4 digital.

Zagalo, Nelson (2009). *Emoções Interactivas, do Cinema para os Videojogos (Interactive Emotions, from Film to Videogames)*. Coimbra: Gracio Editor.

Zagalo, Nelson, Ana Torres, and Vasco Branco (2005). "Emotional Spectrum Developed by Virtual Storytelling". In: *3rd International Conference on Virtual Storytelling*. Strasbourg.

Zimmerman, John, Jodi Forlizzi, and Shelley Evenson (2007). "Research Through Design As a Method for Interaction Design Research in HCI". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '07. ACM, pp. 493–502.

Zimmerman, John, Erik Stolterman, and Jodi Forlizzi (2010). "An Analysis and Critique of Research Through Design: Towards a Formalization of a Research Approach". In: *Proceedings of the 8th ACM Conference on Designing Interactive Systems*. DIS '10. ACM, pp. 310–319.

Acronyms

AGE Authorial Game Evolution Tool. 4, 113

DSR Design Science Research. 77

EDPCG Experience Driven Content Generation. 2, 36, 44

GA Genetic Algorithm. 132

PCG Procedural Content Generation. 2, 36

Glossary

Archetype Game Prototype that serves as the basis for AGE's generative evolution; previous name for the Base-game concept. 113, 114

Authorial Game Evolution A tool and approach for author-led optimization of player-experience. 4, 113

Base-game Game Prototype that serves as the basis for AGE's generative evolution. 160

Design Goal Experience indicator test, in a IF-THEN-ELSE format that expresses designer intent. 160

Design Problem A set of Game Variations and Design Goal tests for a given game; as the name implies, codifies a single problem designers wish to solve with the AGE tool. 162

Experience Indicator Quantitative Indicator of some quality of player experience. 113

Experience-Driven Procedural Content Generation Generic and effective approach for the optimization of player experience. 36

Game Candidate AGE generated solution. 165

Game Feature Content parameter that AGE varies to generate new content; previous name for the Game Parameter concept. 113, 114

Game Parameter Content parameter that AGE varies to generate new content. 165

Game Permutation AGE generated solution; previous name for the Game Candidate concept. 160

Game Variation Definition of how a Game Parameter should be varied with AGE. 160

Game-play Metrics Records of user interaction during game play. 20

Genetic Algorithm Biologically inspired search algorithm; used by AGE in experiments. 132, 136

Procedural Content Generation Automatic or semi-automatic creation of game content. 2, 36

Self-report Subject replies on their perception of player-experience. 18

Target Experience Indicator Designer defined target values for a given experience indicator; previous name for the Target Experience Quality concept. 113, 115

Target Experience Quality Designer defined target values for a given experience indicator. 161