Exploring the Design-Space: The Authorial Game Evolution Tool Case-Study

Rui Craveirinha
Universidade de Coimbra
Coimbra, Portugal
rui.craveirinha@gmail.com

Licinio Roque
Universidade de Coimbra
Coimbra, Portugal
lir@dei.uc.pt

ABSTRACT
This paper describes a case-study on the use of AGE, the Authorial Game Evolution approach, a creativity support tool designed to assist game designers. AGE allows designers to conduct a systematic process of generation and evaluation of game-prototypes, as well as automatically evolve a game-prototype until it mediates a desired form of game-play experience. To assess the tool, a design case study was held where a designer used AGE to create a game. We used a convergent mixed methods experimental design, and analysed quantitative and qualitative data resulting from four design sessions. Creativity Support Index self-report shows the designer found AGE very good in supporting his design, especially for exploration of the design-space. However, he appropriated it exclusively for exploration, not optimization. These show AGE has potential for exploring the design-space, though issues remain before it is an effective medium for high-quality designs.

ACM Classification Keywords
H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

Author Keywords
Procedural Content Generation, Game Design, Taxonomy

INTRODUCTION
One issue in current game design practices is that video-games typically incur in an extensive process of iterative development and evaluation, where prototypes are repeatedly refined, tested in various ways, and tuned [8, 9]. We think that this focus (and consequent effort and cost), arises due to specificities of this medium. 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. As [1] argues, *"the complex nature of [video-game] simulations is such that a result can't be predicted beforehand"*; likewise [7] 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"*. Thus, game designers, in early stages of the design process, must experiment a great number of prototypes, and evaluate them in search of a particular player experience they wish to mediate.

ACE2016, November 09 - 12, 2016, Osaka, Japan

ISBN 978-1-4503-4773-0/16/11…$15.00
DOI: http://dx.doi.org/10.1145/3001773.3001778

Our hypothesis is that if one could turn the process of iterative refinement of video-game prototypes into a semi-automatic design process, where computational tools could take some of the work burden, then it would be possible to streamline video-game design and production, achieving a leaner and more efficient process, and easing the difficulty in finding game-prototypes that achieve designers' intended player-experience.

To achieve this goal, we turned to procedural content generation algorithms or PCG. PCG is method occasionally used in commercial practices for the creation of game content (level design, art assets, AI-controlled characters) by automatic or semi-automatic algorithmic means [6]. Advanced PCG algorithms have been employed to optimize player-experience [14], and help designers in their creative work; for example, by presenting level designers computer-generated alternatives to their current work [13] or filling in low-level detail based on a high-level blueprint [12]. Thus, we aim to incorporate PCG algorithms into game design processes, by framing them in a tool that, in an early stage of the game design process, designers can use it to explore the design-space, and evolve their existing prototype until one is found that meets their creative agenda.

The remaining paper is thus structured. Section 2 details the proposed tool's design and approach, and 3 details the Design Case Study, its set-up, collected data, and observed results. Finally, in section 4, conclusions are discussed.

THE AUTHORIAL GAME EVOLUTION APPROACH
Our proposal is for a Creative Support Tool [10] for game designers that incorporates procedural content generation algorithms. Its goal is to aid developing of video-games that effectively mediate a target player experience, by providing an accessible, semi-automatic process for game designers to iteratively develop, evaluate and fine-tune their video-game prototypes, until their design agenda is achieved.

The tool's underlying approach is based on a previous proposal arguing for an Author-centric Approach to Procedural Content

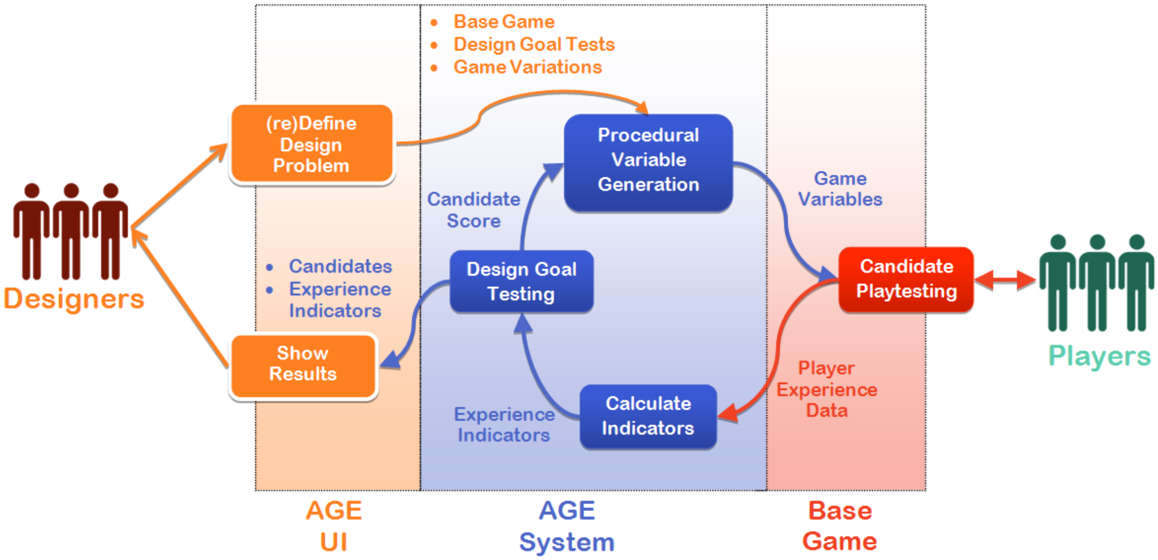

Figure 1. Diagram of the steps in the AGE-powered design process. See section 2 for further details.

Generation [5, 4, 3]. In this work, the design for a customizable form of PCG is outlined and tested on a functional level, with a mock design problem. To actually test its usefulness, data from actual design sessions is needed. With this in mind, we developed a prototype tool and user-interface that could materialize this PCG approach, following guidelines for creative support tools [11]. As per results from [4], it has two major use-cases: providing a designers systematic process of prototype-deployment and player experience inspection, and procedural optimization of a game-prototype.

The resulting tool was named the Authorial Game Evolution tool (AGE for short), for it works by taking an existing base-game prototype, and through its coupling with a procedural content generation algorithm, permits its iterative improvement until the end-result fulfils designer's (authorial) agenda for players' experience. It is a general-purpose tool, theoretically applicable to a vast array of game genres and design problems, and contexts of use.

The gist of how it works is simple: designers create a base-game 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. After configuring the procedural generator, they define a series of tests that evaluate whether or not that prototype is meeting some standard for player-experience that the designer wishes for. Finally, they deploy a series of play-test sessions where players play variations of the original prototype. Data from these sessions is automatically evaluated based on the designer tests; the process repeats in several cycles, until a prototype passes these tests. A more detailed description follows in the next section.

Detailed Description

In terms of how the design process is carried out when in contact with AGE, figure 1 shows how it works step by step. To use the approach, the design and production team must provide 3 main items: a **Base-Game**, a set of **Design Goal Tests**, and a definition of **Game Variations** intended to evolve the base Game. These 3 elements are that which determines the overall specification of the design problem they wish to solve in what regards to their game. Succinctly, the **Base Game** is the prototype that designers wish to evolve; **Design Goal Tests** are a materialization of designers' player-experience aims, and **Game Variations** represent all the exploration possibilities the PCG-algorithm can try so as to achieve them.

The **Base-Game** is a prototype, implemented and be integrated with AGE in a way that leaves several key aspects of its design open to evolution. For example, if a designer wishes to determine the optimal placement of power-up items in a given level, the base game would be the game with all its normal components but without pre-fixed item placing. Instead, it should include a method for placing said items by interacting with AGE.

The designer can use the AGE interface to define a set of **Game Variations**, essentially, the search-space for each **Game Variable** that the designer is looking to improve in their game. In terms of the previous example, this means defining the range of possible item placements. What the PCG algorithm will then do, is generate new values for these variables, and later these will be forwarded to the base-game that will then have to use these to generate the end-object. When players then experience the game, they will play a variation of the **Base Game**, in which those variables were evolved accordingly.

Finally, designers must establish a set of **Design Goal Tests**. A **Design Goal Test** is an attribution of a quality score (that can be positive or negative), to every **Candidate** solution whose player experience indicators pass a given test condition. Experience indicators can be calculated based on data from different sources: game-play metrics, automated subject questionnaires or bio-metrics signals (all that is required is for data to be fed into the AGE system by way of a standardized computational

for all $c \leftarrow Candidates$ **do**
  for all $s \leftarrow PlaySessions_c$ **do**
    $indicator_{c,s} \leftarrow f_i(PlaySessionData_{c,s})$
    if $(Min \leq indicator_c \leq Max)$ **then**
      $score_{c,s} \leftarrow BaseScore * f_s(indicator_c)$
    else
      $score_{c,s} \leftarrow BasePenalty * f_p(indicator_c)$
    end if
  end for
  $score_c \leftarrow average(score_{c,s})$
end for

Figure 2. Algorithmic basis for how AGE evaluates each candidate game solution. Besides getting to define all constants (the indicator test boundaries, *min* and *max*, the *BaseScore* and *BasePenalty*), the UI allows designers to model functions f_i (for calculating indicators based on standard formatted game-play data), and select which types of function are used for score and penalty calculation (f_s,f_p) from a set of standardized functions.

interface). It can then be processed into Experience indicators and designers can model them by inserting mathematical formulae via a UI designed for just that purpose. Furthermore, the UI enables designers to model both the test conditions and how score is attributed when the condition holds (or not).

Design Goal Tests serve as the basis for the procedural algorithm's evaluation of existing prototypes, measuring how close each is to mediating the intended player experience, or in other words, solving the design problem. This is a crucial step in powering the evolution of existing Candidates. In the previous example, let us imagine the designer wished players would collect at least 50% of the in-game power-up items. He could then write a test that measures if in each **Candidate** solution players collect at least half of all items, in which case they would get attributed a positive score. Figure 2.1 shows in detail how candidates are evaluated with **Design Goal Tests**.

The aforementioned three main components, in essence, are the definition of the design problem which the designer seeks to solve. The procedural content generation system will then then try to solve this problem by optimizing **Game Variables**. In other words, these elements codify both the search-space, i.e., the space of all possible design solutions, and the quality evaluation method that translates designers' ideal for players' experience. It is not the intent of this work to make any contribution in terms of the actual procedural optimization algorithms; as such, we use a simple genetic algorithm (GA) to search for the solutions to the design problem; more complex and advanced methods can be easily integrated in its substitution.

Once designers configure AGE via its interface, and it is integrated with the base game, the evolution process begins. The PCG algorithm starts with a phase of candidate solution generation, generating new Game Variable values with the Genetic Algorithm. Then, a play-testing phase ensues: before players play the base-game, it pulls procedurally generated variables from the AGE Tool and employs these as basis for generating the game-content players will experience. As they play with alternative candidates, play-test data is published in the AGE tool. Once enough data has been compiled, it is processed into the Experience Indicators that designers defined. Their values are then tested according to the **Design Goal Tests**, and each candidate is evaluated. As the process cyclically repeats itself until a user-defined terminating condition is achieved (until a given score or maximum number of iterations are achieved), designers access a results pane which allows visualization of all candidate and player experience data, in both table and chart form.

A GAME DESIGN CASE STUDY

Given the complexity of the processes and information required to use this tool, and seeking to validate its usefulness as a Creative Support Tool, AGE needed to be tested in a design context. To do this, we took an AGE prototype and prepared a case study to test how designers worked with this approach, in a scenario as close to real life as possible.

To provide evidence of the approach's validity as a tool for support of the design activity, we wanted to find data that could help answer the following research questions: Can designers define design problems using this approach? How and with what limitations? And also, does this approach help solve design problems in a way that satisfies the designer?

Context

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. Then we asked a game designer to use this 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 games and interactive entertainment applications, and his background mixes computer engineering with photography and design.

The rules we gave him were simple: use AGE to evolve the Dune2 base-game; any game-design agenda would be valid, as long as it could be feasibly incorporated into the base-game. Were this a real game design scenario, the base-game and its integration with the AGE system would be implemented by the designer himself, yet this was not an 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 Variables 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 occurred: 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. 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 subject on his decisions, asking for rationalization for his actions. 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.

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 of the issues he was encountering.

Quantitative data includes user metrics that tracked all designer-interactions with the AGE interface. and a Creativity Support Index [2] report filled by the designer at the end of the final design session, 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 out-loud 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 speech analysis was done, 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 the normal contact-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.

Results

Table 3.3 shows major design events, registered both from the interface and after textual analysis of the designer's dialogue.

From his speech we extracted the following categories. Base-Game Edits are the number of communicated design proposals made by the designer so as to change the game prototype. Cut base-Game Edits refer to edits which the designer expressed verbally but eventually dropped from the existing design. New Game Variables are the number of times the designer asked for a Game Variable to be added to the game prototype for posterior variation. New Metric – number of times the designer asked for a new event during play-testing to be registered by AGE. New Tool Specification – the designer (though we were not expecting it) asked for changes in the workings of AGE, these were counted. For matters of comparison we added some user-interaction events. New Game Variations, refers to every new definition of a game variation to be used by AGE to create new candidates. Edited game Variations – how many times existing Game Variations were changed. New Design Goals for how many design goals were created throughout the exercise and Edited Design Goals for number of edits.

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 Variables	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

Table 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 3.3).

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 3 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. From these, 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 – 1h46m27s and 2h19m45s respectively. In both cases, data indicates greater effort was spent in analysing results as opposed to defining the design. From this wealth of data, we interpreted 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. Note that at neither point, did the designer use AGE to automatically evolve his Design Problems.

The third chart (figure 4) shows a more time-detailed picture within each session. All interface interactions were accounted

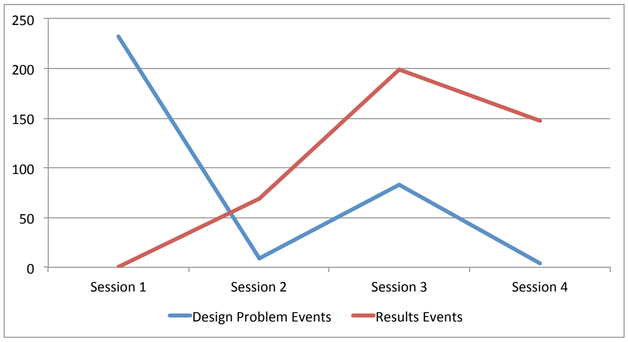

Figure 3. Aggregate of user interactions with the AGE tool interface.

for; however, because these total more than 100 different interface metrics, we needed to select a shorter set for visualization. The chart shows all metrics that had a count value (across the entirety of the experiment) equal to or surpassing 10. In it, we can 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 play-test sessions as a basis for alteration of his 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 25minute 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 table 3.3. 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 can only use it 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[2]). That said, 89 out of a possible 100 points, paints a flattering picture of the tool. It means he found AGE, in this particular case, a 'B+', or 'very good'. Furthermore, it highlights he found it 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 gives credence to 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 from a divergent phase of creative exploration (never trying to converge in on an intended design).

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)		

Table 2. 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.

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.

Session 0 – A Design Brief

The designer's design brief was expressed in a brief written note prepared 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*

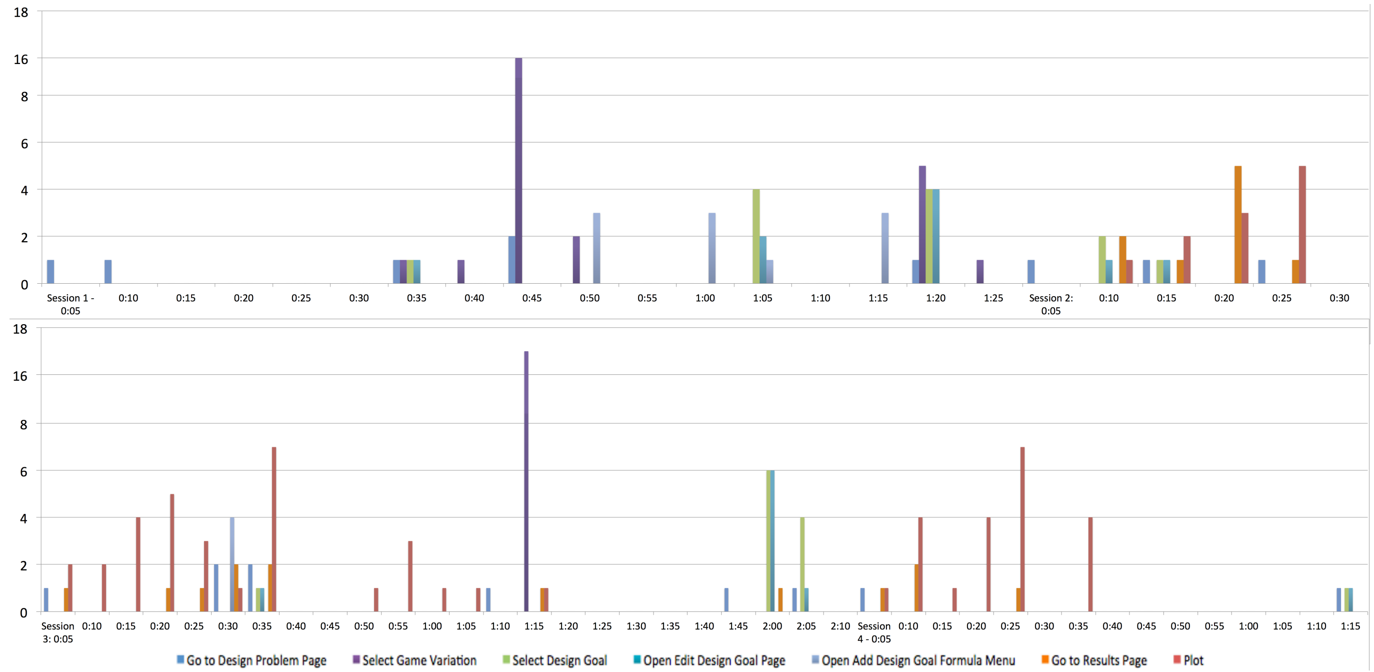

Figure 4. Counts of all UI tool interactions (with a count value equal to or greater than 10) across time, for Sessions 1 and 2 (top) and 3 and 4 (bottom). In blueish and greenish tones are interface events related to defining the design problem. Several moments in these sessions did not have any interaction with AGE, and only involved the designer pondering and expressing his design and results with experimenters. In reddish and orange tones are events related to visualizing results from gameplay testing. Note the vertical scale has been manipulated for issues of space.

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 Variables 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, new data that needed to be present in the game-play metrics extraction system, referring to each units' status (whether moving, stationary, attacking, etc). This was required to be able to capture indicators capable of measuring 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.

Session 1 – Materializing the Brief in AGE

A design phase using AGE ensued where the designer further defined his Design Problem. He started by creating Game Variations, deciding between which values variables would be varied; when asked why he chose those, he replied *"I defined this with minima and maxima [referring to Game Variables' 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.

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 experiment. In this specific use

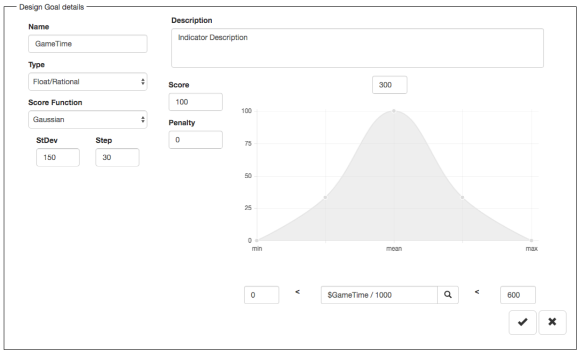

Figure 5. Screenshot of one of the Designers' Design Goals. In this screen, users define how AGE should score and evaluate a given candidate Solution, and therefore optimizes generated variants. Users can edit an indicator formula, select boundaries of values which get positive scores, and even model the function that attributes the score.

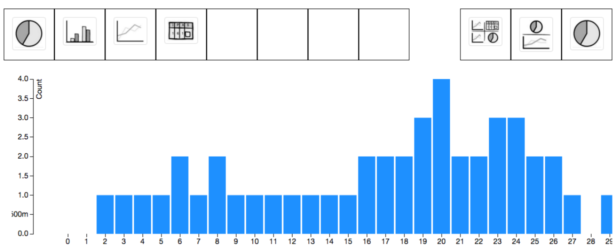

Figure 6. Screenshot of the results pane where Designers can inspect what occurred in playtesting sessions.

case, the system uses random Game Variable 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"*.

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 into 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 of translating 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.

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 though clearly not state as a goal (players were told to collect X resources at the beginning), had a mission success screen. The game *"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 discussed out-loud 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 table 3.3), but in the end he was content with 3 changes. The first was that 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 help him interpret the game's 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.

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 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 all the time avoiding losing all units and avoiding destroying all of the opponents units. He then deployed another round of play-testing, to see if the effect was as intended.

Thus, he concluded the four sessions without terminating his design process, still in an active search for a game that was in accord with his brief.

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, as its interaction"*; to him, this was *"a normal difficulty, that of learning [how to use] a new tool"*. We can, for the most part, disregard this first issue as learning is a necessary evil, one that while possible to alleviate (through an improved interface) can perhaps never be completely solved, and that has 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 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 he struggled. This results as an issue of 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, and in our opinion, the more relevant, 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 kept thinking 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 prevision of how the good the game would end up being was an *"8 or 9, because I've had the possibility of* ***seing*** *people play the game and I think it is fulfilling the... [goals]"*.

Prompted to compare designing with AGE with his traditional design process, 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 the 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 games 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 key word 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"*.

DISCUSSION

Even after the 5 sessions, the designer never got to a point of trying to evolve his game, nor did he find the right game design that fulfilled his experience agenda; these are both undesirable results as the AGE tool was created to help designers improve mediated player experience. This may be a hint that there is an intrinsic problem with this approach.

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 what design's search-space was an intuitive process. However, despite early definition of Design Goals, he did not use with to the effect of evolving his base game automatically. But we find evidence 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 encountered 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 abstract concepts? Twice, he went as far as playing with values in google's calculator, in a visible effort that led to unfruitful results. Even his proposals for new data sources and charts did not solve this.

So he could verbally describe his desired form of player experience, but could not translate it into a set of objective, measurable and verifiable conditions, that used player behaviour as a basis, which is what AGE requires to work. His main answer 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 grasped yet the existing game prototype and player-experience. But the uncertainty is there: can designers easily transport their vision of experience into a set of objective tests that automatically evaluate games? In other words, **Is AGE's Design Problem metaphor usable by designers?** Also, he questioned the very nature of this game and its adequateness to AGE. Thus, **In what design cases can AGE be used effectively?**

Note that this is not a completely negative result: the designer was clearly happy with how the design was evolving – both the CSI and post-experiment interview show 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. We also found several conclusions that we feel are of interest to the community.

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 sub-optimal configurations that can latter be iteratively improved. 3 strands of evidence confirm this: one, the designer himself in his interventions (especially the final interview), two, the nature of the design process, where PCG was used only for exploration purposes, and three, the CSI questionnaire, where the highest score was in the exploration category. This is a somewhat unexpected value found in a tool that uses PCG, as there is a great deal of research focused on optimization (see [14] for instance). This suggests 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 deployment and data collection of game prototypes – a semi-automatic emulation of traditional design processes – and accompanying mindset it suggested, was valued by the designer in of itself, despite any possible gains in effort permitted by PCG. Then perhaps PCG was not even the greatest addition of value in the tool. Hence, **Tools that systematize development, evaluation and testing of prototypes can aid game designers**.

Despite focus on exploration, the designer affirmed that in the context of past game designs of his, he would definitely use 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 an **Exploration phase before an Optimization phase**.

Even though the designer struggled with defining 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 would surely do so continuously until he found a good solution. AGE assisted by providing an easier way to explore several different design alternatives simultaneously, a structured work flow, and an application that eased the process of 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.

Concluding, 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, naturally, it serves only to give rise to research questions which empirical studies can later prove or disprove.

ACKNOWLEDGMENTS
This work was financed under PhD Scholarship number SFRH/BD/75196/2010 awarded by the FCT - Fundação para as Ciências e Tecnologia.

REFERENCES

1. Espen Arseth. 2001. Computer Game Studies, Year One (editorial). In *the international journal of computer game research*, Espen Arseth (Ed.). Vol. 1. Issue 1.

2. Erin Cherry and Celine Latulipe. 2014. Quantifying the Creativity Support of Digital Tools Through the Creativity Support Index. *ACM Trans. Comput.-Hum. Interact.* 21, 4 (2014).

3. 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*.

4. 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*. 383–390.

5. R. Craveirinha, L.L. Santos, and Licinio Roque. 2013. An Author-Centric Approach to Procedural Content Generation. In *ACE 2013 the 10th International Conference in Advances in Computer Entertainment Technology*.

6. Mark Hendrikx, Sebastiaan Meijer, Joeri Van Der Velden, and Alexandru Iosup. 2013. Procedural Content Generation for Games: A Survey. *ACM Trans. Multimedia Comput. Commun. Appl.* 9, 1 (2013), 1:1–1:22.
7. Robin Hunicke, 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.
8. Dan Irish. 2005. *The Game Producer's Handbook*. Course Technology Press, Boston, MA, United States.
9. Clinton Keith. 2010. *Agile Game Development with Scrum* (1st ed.). Addison-Wesley Professional.
10. Ben Shneiderman. 2007. Creativity Support Tools: Accelerating Discovery and Innovation. *Commun. ACM* 50, 12 (Dec. 2007), 20–32.
11. Ben Shneiderman, Gerhard Fischer, Mary Czerwinski, Mitch Resnick, Brad Myers, Linda Candy, Ernest Edmonds, Mike Eisenberg, Elisa Giaccardi, Tom Hewett, Pamela Jennings, Bill Kules, Kumiyo Nakakoji, Jay Nunamaker, Randy Pausch, Ted Selker, Elisabeth Sylvan, and Michael Terry. 2006. Creativity Support Tools: Report From a U.S. National Science Foundation Sponsored Workshop. *International Journal of Human-Computer Interaction* 20, 2 (2006).
12. Gillian Smith, Jim Whitehead, and Michael Mateas. 2010. Tanagra: a mixed-initiative level design tool. In *Proceedings of the Fifth International Conference on the Foundations of Digital Games (FDG '10)*. ACM, 209–216.
13. Georgios N Yannakakis, Antonios Liapis, and Constantine Alexopoulos. 2014. Mixed-initiative co-creativity. In *Proceedings of the ACM Conference on Foundations of Digital Games*.
14. Georgios N. Yannakakis and Julian Togelius. 2011. Experience-Driven Procedural Content Generation. *IEEE Transactions on Affective Computing* 99 (2011).