An Experiment on Content Generation of Game Software Engineering

Anonymous Author(s)

ABSTRACT

Background Video games are complex projects that involve a seamless integration of art and software during the development process to compound the final product. In the creation of a video game, software is fundamental as it governs the behavior and attributes that shape the player's experience within the game. When assessing the quality of a video game, one needs to consider specific quality aspects, namely 'design', 'difficulty', 'fun', and 'immersibility', which are not considered for traditional software. On the other hand, there are not well- established best practice for the empirical assessment of video game as instead there are for the empirical evaluation of more traditional software. **Aims** Our goal is to carry out a rigorous empirical evaluation of the latest proposals to automatically generate content for videogames following best practise established for traditional software. Specifically, we compare Procedural Content Generation (PCG) and Reuse-based Content Generation (RCG). Our study also considers the perception of players and professional developers on the content generation. **Method** We conducted a controlled experiment where human-subjects had to play with and evaluate content automatically generated for a commercial video-game by the two techniques (PCG and RCG) based on specific quality aspects of video games. 44 subjects including professional developers and players participated in our experiment. **Results** The results suggest that RCG generates content of higher quality than PCG which is more aligned with the pre-existent content. **Conclusions** The results can turn the tides for content generation. RCG has been underexplored so far because the reuse factor of RCG is perceived as repetition by the developers, who ultimately want to avoid repetition in their video games as much as possible. However, our study revealed that using RCG unlocks latent content that is actually favoured by players and developers.

KEYWORDS

Empirical Study, Automated Software Transplantation, Procedural Content Generation

ACM Reference Format:
Anonymous Author(s). 2024. An Experiment on Content Generation of Game Software Engineering. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM. . . $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Video games industry is in continuous growth every year [43]. Despite being one of the fastest growing industries, video game software engineering has been identified as an area of knowledge that needs more fundamental research [2, 13]. One of the areas where video game software engineering needs more rigorous research are empirical research methods [13].

While theoretical frameworks provide foundational understanding, empirical studies offer the necessary validation and refinement crucial for effective implementation. As in other disciplines dealing with human behaviour (e.g., social sciences or psychology), empirical research allows building a reliable knowledge base in software engineering [49, 58]. By empirically investigating the user experience of video game techniques, researchers can illuminate both the strengths and limitations of existing approaches, paving the way for advancements that align more closely with the diverse needs and preferences of developers and players.

One of the video game development challenges is the need of content [51]. Content generation is often a slow, laborious, costly, and error-prone process. This results in issues such as significant delays in content development [33, 56] and the growing need for game content from post-launch updates. Through rigorous experimentation, empirical studies can serve as the cornerstone for pushing the boundaries of what is achievable within content generation.

In this work, we aim to evaluate empirically two different video game content generation techniques along with two different users profiles (players and developers). We study the feasibility of Procedural Content Generation (PCG) and Reuse-based Content Generation (RCG), and whether they have an impact on the quality of the generated content. We do so by analyzing Kromaia, a commercial video game released on PlayStation 4 and Steam.

We present an experiment in which we compare content generated by RCG and PCG, in terms of video game specific measures 'design', 'difficulty', 'fun', and 'immersibility'. A total of 44 subjects performed the tasks of the experiment, assessing the generated content in two scenarios of Kromaia. We conduct three distinct sessions, one for players and the other two for developers, in order to investigate whether the profile of the participants assessing video games influences their perception.

Our results suggest that the two techniques compared in the experiment, RCG and PCG, result in bosses with different quality perceived by the subjects. The results show that the subjects perceive the boss generated by RCG to be of superior quality in comparison to the one generated with PCG. In addition, we observe no statistical difference when assessing the quality of bosses based on the subject profile (players and developers). However, we find statistical significant differences in the comments made by the subjects and in the won rate (the amount to of times a player can win against a boss).

This study presents a new perception when reusing content. Developers perceived reusing as repetitive but is far from our results which clearly shows an opportunity for content generation. We also notice that the subjects of the experiment does not need to be limited to players as have been done until now. On other hand, following with the traditional software engineering good practices, we provide a replication package for future researchers of the game software engineering community.

By merging the conventional methods of software experimentation with the specific attributes of video games, our project has uncovered new prospects for advancing game software engineering. Additionally, our research has broadened the horizons for content generation utilizing RCG possibilities, as well as diversified the subjects for evaluating the generated content.

The structure of this paper is as follows. Section 2 reviews the related works in the area. Section 3 presents the approaches under study and the context of the experiment, Kromaia. Section 4 outlines the experimental design. Section 5 presents the experiment results, followed by a discussion in Section 6. Section 7 summarizes the threats to the validity. Finally, Section 8 concludes the paper.

2 RELATED WORK

Experimentation in software engineering is a practice that has been studied for decades [6]. Throughout time, researches have adopted established guidelines to be rigorous [58], such as the use of hypothesis, validity, statistical analysis or replication packages.

Content generation is a large field [59]. The types of content generated are diverse, such as vegetation [34], sound [39], terrain [22], Non-Playable Characters [55], dungeons [54], puzzles [17], and even the rules of a game [10]. However, it is difficult to find experiments with human-subjects that compare approaches [3].

Table 1 shows content generation work with human-subjects. In content generation, it is common that experiments with human subjects explore the quality of the generated content [9, 50] or different variants of the proposed approach [1, 38]. On other hand, work such us Pereira *et al.* [37] or Prasetya *et al.* [42] compared the generated content by their approach to a baseline (see Evaluation column of Table 1). In this work, we compare two techniques for generating content that the community uses without any previous experiments to compare them.

In terms of measurements, studies have been conducted to examine the distinctive characteristics of video games [44]. Studies have investigated subjects, more precisely players, preferences and perceptions regarding various aspects of video games, including design [28, 36], difficulty [32, 37], or fun [39, 42]. Another aspect of video games is the user engagement and immersion, which plays crucial roles in shaping the overall gaming experience [27] (see Measurements column of Table 1). Our work considers all these measurements simultaneously.

Table 1 shows that none of the previous work is compliant with the practices adopted in experiments by traditional software. In fact, 65% have neither hypothesis and validity, statistical analysis nor replication package (see Hypothesis & Validity, Statistical Analysis, and Replication Package columns of Table 1). Our work aims to compare with empirical rigour the content generated. To do so, we adopted traditional software guidelines for experimentation.

Table 1: Overview of related work. Evaluation: generated content (A), variants of the proposed algorithm (VA), generated content compared to a baseline (C). Measurements: Design (De), Difficulty (Diff), Fun (F), Human Made (HM), Immersibility (I).

Work Year	Evaluation	Measurements	Hypothesis & Validity	Statistical Analysis	Replication Package	Sample
Cardamone *et al.* [11] 2011	VA	De	✗	✗	✗	5 players
Plans *et al.* [39] 2012	A	F	✗	✓	✗	31 players
Adrian *et al.* [1] 2013	VA	De, Diff, F	✗	✗	✗	22 players
Dahlskog *et al.* [16] 2013	VA	De, Diff, F	✗	✗	✗	24 players
Togelius *et al.* [50] 2013	A	De, Diff, F	✓	✓	✗	147 players
Gravina *et al.* [24] 2015	A	F	✗	✗	✗	35 players
Kaidan *et al.* [28] 2015	VA	De	✗	✗	✗	12 players
Olsted *et al.* [36] 2015	VA	De	✗	✗	✗	13 players
Prasetya *et al.* [42] 2016	C	F	✗	✗	✗	33 players
Ferreira *et al.* [20] 2017	VA	De, Diff, F, I	✗	✓	✗	139 players
Charity *et al.* [12] 2020	A	De, Diff	✗	✗	✗	2 players
Lopez-Rodriguez *et al.* [32] 2020	VA	Diff	✗	✗	✗	30 players
Kraner *et al.* [30] 2021	A	De	✗	✗	✗	5 players
Pereira *et al.* [37] 2021	VA	Diff, F, HM	✗	✓	✗	70 players
Pereira *et al.* [38] 2021	C	Diff, F	✗	✓	✗	16 players
Brown *et al.* [9] 2022	A	De	✗	✗	✗	35 players
De Lima *et al.* [18] 2022	A	HM	✗	✓	✗	38 players
Our work	PCG vs RCG	De, Diff, F, I	✓	✓	✓	32 players + 12 developers

Thus far, previous work has only used players to evaluate content. In other words, they have not considered the perception of the developers themselves (see Sample column of Table 1). We study not only the players assessment, but also the point of view of professional video game developers, and their differences when assessing the quality of the generated content.

3 BACKGROUND

In this section, we present the importance of software in video game development, the generation of content for video games, and the real-world context that we make use of on our experiment to perform the corresponding tasks.

3.1 Software in video games

The development process of video games requires a harmonious combination of artistic elements and software integration, resulting in intricate and multifaceted creations. Software plays a crucial role in every aspect of a video game's creation as it dictates the behavior and features that can be seen or experienced within the game. For instance, software is responsible for controlling the logic behind the behaviors of non-playable characters (NPCs) in a game. As video games evolve and become more sophisticated, the software powering them also becomes increasingly intricate.

Nowadays, most video games are developed by means of game engines. One can argue that game engines are software frameworks [41]. Game engines integrate a graphics engine and a physics engine as well as tools for both to accelerate development. The most popular ones are Unity and Unreal Engine, but it is also possible for a studio to make its own specific engine (e.g., CryEngine [15]).

One key artefact of game engines are software models. These are software models such as those proposed by the Model Driven Development paradigm [46] which should not be confused with either 3D Meshes or AI Models. Unreal proposes Unreal Blueprints [8], Unity proposes Unity Visual Scripting [45], and a recent survey in Model-Driven Game Development [60] reveals that UML and Domain Specific Language (DSL) models are also being adopted by development teams. Developers can use the software models to create video game content instead of using the traditional coding approach (C++ on Unreal or C# on Unity). While code allows for more control over the content, software models raise the abstraction level, thus promoting the use of domain concepts and minimizing implementation and technological details.

3.2 Content Generation for Video Games

The process of content generation for video games is typically slow, tedious, expensive, and susceptible to errors. Thus, leading to problems that the industry have such us: (1) excessive delays in content creation (with notorious examples in Cyberpunk 2077 [56] or GTA VI [33]) or (2) the ever-increasing demand for game content derived from post-launch updates, Downloadable Content (DLCs), games as a service, or platform-exclusive content.

To address these challenges, researchers have been exploring procedural content generation techniques as a potential solution to (semi)automate the generation of new content within video games [26]. Procedural content generation can be grouped in three main categories according to the survey by Barriga *et al.* [5]: Traditional techniques that generate content under a procedure without evaluation; Machine Learning techniques [31, 47] that train models to generate new content; and Search-Based techniques [52] that generate content through a search on a predefined space guided by a meta-heuristic using one or more objective functions.

Content can also be created through reuse. In fact, since the term software engineering was coined at the NATO Conference held in Garmisch in 1968 [35], its evolution has been tied to the concept of reuse. Either applying an opportunistic approach such as clone-and-own [21], or applying systematic approaches as software product lines (assembling predefined features) [40] or as software transplantation (a feature is transplanted from a donor to a host) [4]. A recent SLR on game software engineering [13] identifies the relevance of both Procedural Content Generation (PCG) and Reuse-based Content Generation (RCG).

3.3 Kromaia Video Game for the Experiment

Kromaia is a commercial video game released on Playstation and Steam, translated into eight languages. On Kromaia, each level consists of a three-dimensional space where a player-controlled spaceship has to fly from a starting point to a target destination, reaching the goal before being destroyed. The gameplay experience involves exploring floating structures, avoiding asteroids, and finding items along the route, while basic enemies try to damage the spaceship by firing projectiles. If the player manages to reach the destination, the ultimate antagonist corresponding to that level (which is referred to as *boss*) appears and must be defeated in order to complete the level.

In the context of Kromaia, developers generate content through PCG by means of the work of Gallota *et al.* (which combines an L-system with an evolutionary Algorithm) [23] because it is specific for spaceships that can play the role of bosses, and it achieves the best state-of-the-art results for this type of content. Developers also generate content through RCG by means of reusing features between Kromaia´s content. Specifically, the developers select a feature (a fragment of content) from a donor, and a host (another content) that will receive the feature. Despite the research efforts in both PCG and RCG and the importance of content generation for video game development, there is no study that directly compares them.

4 EXPERIMENTAL DESIGN

In this section we present the experiment design following the Wohlin's guidelines [58] for reporting software engineering experiments.

4.1 Objectives

The research objective has been organized using the Goal Question Metric template for defining objectives originally presented by Basili and Rombach in their 1988 publication [6].

Our goal is to **analyze** different techniques in content generation: Procedural Content Generation (PCG) and Reuse-based Content Generation (RCG); **for the purpose of** comparison, **with respect to** perceived quality; **from the point of view of** of more and less experienced players and developers; **in the context of** new content generation for an existing video game.

4.2 Research Questions and Hypotheses

The research questions and null hypotheses are formulated as follows:

RQ1 - Does the **Technique** used to automatically generate software in video games impact the perceived *Quality* of the game? The corresponding null hypothesis is $H_{0,1}$: The **Technique** does not have an effect on the perceived *Quality* of the game.

RQ2 - Does the **Evaluator's profile** impact the evaluation of the *Quality* of the game? The corresponding null hypothesis is $H_{0,2}$: The **Evaluator's profile** does not have an effect on the evaluation of the *Quality* of the game.

The hypotheses are formulated as two-tailed hypotheses, as this is the first comparison between the two techniques with subjects.

4.3 Variables

In this study, the factor under investigation is the content generation technique (**Technique**) used for automatically generate elements, final bosses, for an existing video game. There are two alternatives: PCG or RCG, which are the two different techniques used to generate a final boss that will be played with and evaluated by different kind of subjects.

Since the goal of this experiment is to evaluate the effects of using different techniques to generate content for an existing commercial video game, we selected response variables related to the quality perceived by subjects playing the generate content. We decomposed the analysis of quality into different dimensions: design, difficulty, fun and immersibility, based on the measurements used in previous works.

To evaluate difficulty we defined three response variables: *Game duration*, *Won rate* and *Boss difficulty*. We defined *Game duration* as the average time spent by each subject in their games. The value of this variable was calculated by dividing the time each subject spent playing with a boss by the number of games played against that boss. *Won rate* is the percentage of games won by a player out of all games played against a boss, and we calculated it dividing the number of games won by the number of games played against a boss. We measured *Boss difficulty* with the subject's answers to an explicit question about the difficulty of the game in a 7-item Likert-type questionnaire with different items. Different items in this questionnaire were used to measure the response variables *Design*,*Fun*, and

Table 2: Response variables and correspondent items in the evaluation questionnaire

Response variable	Related Items in the evaluation questionnaire
Boss difficulty	*Item*1. I think the boss difficulty is high.
Design	*Item*2. The boss is perfectly integrated in Kromaia *Item*3. I liked the design and behavior of the boss *Item*4. The boss I fought seemed to me to have a good balance between difficulty and playability.
Fun	*Item*5. I enjoyed playing against the boss It6. When the time was up, I was disappointed that I could not continue playing against the boss.
Inmersibility	*Item*7. At no time did I want to give up while facing the boss. *Item*8. At some point I was so involved that I wanted to talk directly to the video game

Immersibility. Each of these variables correspond to specific items in the questionnaire. The subjects rated their degree of agreement with the statements of each item, with a value of 1 corresponding to totally disagree and 7 to totally agree. We average the scores obtained for these items to obtain the value for each variable. Table 2 show the specific items of the questionnaire assigned to the calculation of each of these response variables.

For the evaluation of each boss in the game, the subjects also answered an open-ended question in which they could add comments that could not be taken into account through the questionnaire. We considered two response variables to quantify the qualitative information contained in these comments: *Comment length*, defined from the number of characters in the comment, and *Comment Type*. To define the type of comment, the comments were classified into five categories by assigning them a numerical value from 0 to 4: 0, no comments; 1, comments not related to the evaluation of the boss; 2, comments on the difficulty of the boss evaluated; 3, comparisons between the bosses played; and 4, detailed analysis of the evaluation made.

In order to establish the different evaluator profiles among the participating subjects, we conducted different sessions of the experiment with specific groups of subjects: potential gamers and experienced developers. In addition, a demographic questionnaire was designed to take into account the degree of experience both playing and developing video games, in particular, playing video games with similar characteristics to the one being evaluated. Both the groupings of subjects in sessions by participant profile (player or developer) and the subjects' responses to the demographic questionnaire were used to define three confounding factors: **Profile**, **Game development**, and **Gamer profile**. The objective was to analyze whether and how experience in video game development and profile as a gamer could influence the evaluation of the quality of the elements of the game played.

The factor **Profile** has two alternatives, player or developer, depending on the previous grouping of subjects in sessions by profile. This factor also allows the study of the differences between the sessions held and the demographic profiles of the participating subjects. To define the alternatives for the factor **Game development**, the weekly hours that the subjects dedicated to developing software for video games were taken into account. The factor will have two alternatives: 1, for subjects who do not dedicate more than 10 hours per week to developing video games, and 2 for those who dedicate 10 hours or more to developing video games each week. The **Gamer profile** factor is used to distinguish subjects with a gamer profile that is closer to the target audience of the video game being analyzed from subjects with less related profiles, such as casual gamers or those who are not interested in video games. In order to define the alternatives of the factor **Gamer profile** we considered the scores given by the subjects to the following questions:

1. How many hours do you play video games per week? (1, Less than 5; 2, between 6 and 10; 3 between 11 and 20; 4, between 31 and 30; 5, between 31 and 40; and 6 more than 40.)
2. How would you rate your overall experience with video games (knowledge, playing time, skills)? (1, No experience; 2, Little experience; 3, Medium experience; 4, Very experienced; and 5, Expert in the area)
3. How would you rate your overall experience with shooter video games (Examples: Call Of Duty, Doom, Quake)? (1, No experience; 2, Little experience; 3, Medium experience; 4, Very experienced; and 5, Expert in the area)
4. What difficulty do you usually choose when playing video games? (1, Easy; 2, Normal; 3, Hard; 4, Extreme)

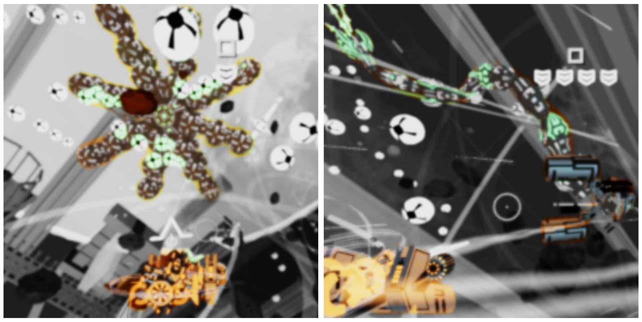
We defined three alternatives for the factor **Gamer profile** according to the sum of the scores given by the subjects to the questions: 1, for subjects scoring no more than 33% of the 20 possible points, 2 for subjects scoring between 33% and 66% of the possible points and 3, for subjects scoring 66% or more of the possible points. Subjects in the third alternative of the factor could be considered the most similar to the target audience of the game, while subjects in the first alternative would represent subjects more distant from this audience.

4.4 Design

We chose a Two-Treament crossover design with two sequences using two different evaluation task: T1, evaluate a game boss made using RCG, and T2, evaluate a game boss made using PCG. The subjects were randomly divided into two groups (G1 and G2). In the first period of the experiment, the subjects of G1 makes T1 and the subjects of G2 makes T2. Afterwards, in the second period, the subjects of G1 makes T2 and the subjects of G2 makes T1.

This repeated measure design enhances the experiment's sensitivity, as noted by Vegas *et al.* [53]. Considering the same subject evaluating both alternatives, between-subject differences are controlled, thus improving the experiment's robustness regarding variation among subjects. By using two different sequences (G1 evaluating RCG first and PCG afterwards, and G2 evaluating PCG first and RCG afterwards) the design counterbalances some of the effects caused by using the alternatives of the factor in a specific order (i.e., learning effect, fatigue). The effects of the factors period, sequence, and subject will be studied to guarantee the validity of this experiment.

To verify the experiment design, we conducted a pilot study with two subjects. The pilot study facilitated an estimate of the time required to complete the tasks and questionnaires, the identification of typographical and semantic errors, and the testing of the online environment used to create the experiment. The subjects in the pilot study did not participate in the experiment.



(a) (b)

Figure 1: (a) PCG boss. (b) RCG boss.

4.5 Participants

We selected the subjects using convenience sampling [58]. A total of 46 subjects with different knowledge about developing and playing video games performed the experiment, but only 44 decided to submit their answers and confirmed their agreement to be part of this study. In this study, the subjects included 12 professionals related with the video game development and 34 third year undergraduate students who are taking a course in *Software Quality* from different technology programs at Universidad San Jorge. In particular, part of those students were studying specifically to design and develop video games.

The experiment was conducted by two instructors. During the experiment, one of the instructors gave instructions and managed the focus groups, and both instructors clarified doubts and took notes.

4.6 Experimental Objects

In the experiment the subjects evaluate specific elements, bosses, created for a existing video game, Kromaia [7]. Subjects must eliminate these bosses by piloting and shooting from a spaceship. Figure 1 shows the spaceship used by the player and the two bosses used during the experiment; Figure 1a shows the boss generated by PCG, and Figure 1b shows the boss generated by RCG. For the execution of this experiment an video game engineer, who was involved in the development of Kromaia developed a test scenario based on scenarios from the original Kromaia game. In this scenario the subjects participating in the experiment can (1) learn how to operate the game controls, (2) fight an original boss from the game, as well as (3) fight different bosses that they will have to evaluate.

For data collection, we prepared two forms using Microsoft Forms (one for each experimental sequence) with the following main sections. The section IV was repeated three times in the questionnaires, once for each boss played by the subjects: first against the original Kromaia boss, and then with the bosses generated with the techniques we compared (PCG and RCG):

I An informed consent form that the subjects must review and accept voluntarily. It clearly explains what the experiment consists of and that the personal data will not be collected.
II A demographic questionnaire that was used for characterizing the sample and defining the confounding factors.
III Specific information on how to download and use the Kromaia test environment that will be used to perform the experiment, and instructions on how to use the game environment.
IV Specific instructions on how to access the boss fight and the evaluation questionnaire about the game experience against the boss.

The experimental objects used in this experiment (the testing Kromaia scenario, the playing bosses, and the forms used for the questionnaires), as well as the results and the statistical analysis, are available as a replication package at http://svit.usj.es/RCGvsPCG. TODO *put replication package anonymous*

4.7 Experimental Procedure

The experiment was conducted in three different sessions. In the first session, the experiment was conducted face-to-face with the group of students. In the second and third session, the experiment was conducted online with professionals. During the online session, all the participants joined the same video conference via Microsoft Teams, and the chat session was used to share information or clarify doubts. The experiment was scheduled to last for one hour and 40 minutes and was conducted following the experimental procedure described as follows:

(1) An instructor explained the context of the experiment, the parts of the session and clarified that the experiment was not a test of the subjects' abilities. (5 min)
(2) The subjects received clear instructions on where to find the links to access the forms for participating in the experiment and about the structure of these forms. The subjects were randomly divided into two groups (G1 and G2). (10 min)
(3) The subjects accessed the online form, and they read and confirmed having read the information about the experiment, the data treatment of their personal information, and the voluntary nature of their participation before accessing the questionnaires and tasks of the experiment. (5 min)
(4) The subjects completed a demographic questionnaire. (5 min)
(5) The Subjects received specific information on how to download and use the Kromaia test environment that will be used to conduct the experiment. They downloaded and used the Kromaia test environment to learn how to pilot the ship they will had to use to fight different bosses during the experiment. (15 min)
(6) The subjects received specific instructions on how to access to a fight with an original boss of Kromaia. After playing against the boss as many times as desired, the subjects completed the evaluation questionnaire about the experience of playing against the original boss. (15 min)
(7) The subjects performed the first task. They received specific instructions on how to access to a fight with the boss to evaluate. The subjects of G1 played against the boss generated with RGC while the subjects of G2 played against the boss generated with PCG. After playing as many times as desired against the assigned boss, all the subjects completed the evaluation questionnaire about the game experience against the boss played. (15 min)
(8) The subjects performed the second task. They received instructions on how to access to a fight with the boss to evaluate.

The subjects of G1 played against the boss generated with PCG while the subjects of G2 played against the boss generated with RCG. After playing as many times as desired against the assigned boss, all the subjects completed the evaluation questionnaire about the game experience against the boss played. (15 min)

(9) One instructor conducted a focus group interview about the tasks, while the other instructor took notes. (15 minutes)

(10) Finally, a researcher analyzed the results.

4.8 Analysis Procedure

We have chosen the Linear Mixed Model (LMM) [57] for the statistical data analysis. LMM handles correlated data resulting from repeated measurements, and it allows us to study the effects of factors that intervene in a crossover design (period, sequence, or subject) and the effects of other confounding factors (e.g., in our experiment, profile, game development practice, and gamer profile) [53]. In the hypothesis testing, we applied the Type III test of fixed effects with unstructured repeated covariance. This test enables LMM to produce the exact F-values and p-values for each dependent variable and each fixed factor.

In this study, **Technique** was defined as a fixed-repeated factor to identify the differences between using PCG or RCG, and the subjects were defined as a random factor $(1|Subj)$ to reflect the repeated measures design. The response variables (RV) for this test were: *Game duration*, *Won rate*, and *Boss difficulty*, *Design*,*Fun*, and *Immersibility*. We also analyzed the response variables *Comment length* and *Comment Type*to determine differences in the comments of the subjects.

In order to take into account the potential effects of factors that intervene in a crossover design in determining the main effect of **Technique**, we considered **Group** to be fixed effect with two alternatives: G1 and G2, corresponding to the two different sequences in which the bosses are evaluated. The first group of subjects (G1) played and evaluated the boss generated with RGC, and then played and evaluated the boss generated with PCG. The second group of subjects (G2) played and evaluated the boss generated with PCG, and then played and evaluated the boss generated with RGC.

In order to explore the potential effects of the confounding factors related to the evaluator's profile to determine the variability in the response variables, in the statistical model we also considered the fixed factors **Profile**, **Game development**, and **Gamer profile** and the combination of this factors with the principal factor **Technique**

We tested different statistical models in order to find out which factors, in addition to **Technique**, could best explain the changes in the response variables. Some of these statistical models are described mathematically in Formula 1. The starting statistical model ($Model\,0$) reflects the main factor used in this experiment, **Technique** (Tech.)and the random factor $(1|Subj)$. We also tested other statistical models (e.g., $Model\,1$, $Model\,2$, and $Model\,3$) that included the one or more of the confounding factors (CF) considered in the experiment (**Group**, **Profile**, **Game development**, or **Gamer profile**) or their interactions with the factor **Technique** ($Tech. * CF$) which could have effects on the response variables.

$$\begin{array}{ll} Model\,0 & RV \sim Tech.+\left(1|Subj.\right) \\ Model\,1 & RV \sim Tech.+CF+Tech.*CF+\left(1|Subj.\right) \\ Model\,2 & RV \sim Tech.+CF_1+CF_2+CF_3+CF_4+\left(1|Subj.\right) \\ Model\,3 & RV \sim Tech.+CF_1+CF_2+Tech.*CF_1+\left(1|Subj.\right) \end{array} \quad (1)$$

The statistical model fit of the tested models for each variable was evaluated based on goodness of fit measures such as Akaike's information criterion (AIC) and Schwarz's Bayesian Information Criterion (BIC). The model with the smallest AIC or BIC is considered to be the best fitting model [19, 29]. The assumption for applying LMM is the normality of the residuals of the response variables. To verify this normality, we used Kolmogorov-Smirnov and Shapiro-Wilk tests as well as visual inspections of the histograms and normal Q-Q plots. To describe the changes in each response variable, we selected the statistical model that satisfied the normality of residuals and also obtained the smallest AIC or BIC value.

To quantify the differences in the dependent variables due to the factors considered, we calculated the Cohen d value [14], which is the standardized difference between the means of the dependent variables for each factor alternative. Values of Cohen d between 0.2 and 0.3 indicate a small effect, values around 0.5 indicate a medium effect and values greater than 0.8 indicate a large effect. We selected box plots to describe the results graphically. To verify that the group of measurements associated with each response variable or factor is consistent, we applied Principal Components Analysis (PCA) to the set of measurements collected from the task sheets. PCA allows analyzing the structure of the correlations in a set of variables, identifying and establishing subsets of variables that have "something" in common with each others, but not with the rest. PCA produces components, which are new random variables that summarize the patterns of each subset of variables and are not correlated with each other [25, 48]. If the set of measures selected to define a variable (e.g., the results of items 2, 3, and 4 to define variable *Design*) are in a single PCA component, the information from the measures is correlated and can be reduced into one variable, which would support the consistency of the proposed grouping of metrics. On the other hand, if the metrics used to define different variables are in different PCA components, we can interpret that they explain different aspects of the information contained in the measures and that there is no strong correlation between them.

5 RESULTS

5.1 Changes in the response variables

There were differences in the means and standard deviations of all of the response variables related with the boss quality perceived by the subjects depending on which **Technique** was used to create the played boss. However, the differences in *immersibility* were small and there were also no large differences due to the factor **Technique** in the variables related to the subjects' comments. Table 3 shows the values for the mean and standard deviation of all the response variables considered *Game duration*, *Won rate*, *Fun*, *Boss difficulty*, *Design*, *Fun*, *Immersibility*, *Comment length*), and *Comment Type* for each one of the **Techniques** compared: PCG and RCG, and for each one of the alternatives of the confounding factors considered as fixed factors in the statistical analysis: **Profile**, with two alternatives

Table 3: Number of cases and Values for the mean and standard deviation ($\mu \pm \sigma$) of the dependent variables for the factor (Technique) in each alternative of the fixed factors

		Technique	Profile		Developing Games		Gamer Profile			Group	
			Players	Developers	More than 10 h/Week	Less than 10 h/Week	Target Audience	Neutral	Non Target Audience	G1 (RCG-PCG)	G2 (PCG-RCG)
Game Duration	RCG	4.24±2.85	4.18±3.23	4.38±1.52	4.05±3.27	4.57±1.95	4.57±4.36	3.22±2.22	5.33±2.77	4.16±2.93	4.32±2.83
	PCG	2.01±1.76	2.19±2.02	1.54±0.55	2.39±2.06	1.34±0.68	1.58±0.54	2.01±1.38	2.13±2.34	2.21±2.28	1.79±0.93
	All	3.12±2.61	3.18±2.85	2.96±1.83	3.22±2.83	2.95±2.18	3.07±3.33	2.62±1.92	3.73±3	3.19±2.77	3.05±2.44
Won rate	RCG	0.32±0.37	0.33±0.39	0.29±0.33	0.3±0.39	0.36±0.35	0±0	0.25±0.32	0.5±0.39	0.41±0.38	0.22±0.34
	PCG	0.71±0.39	0.7±0.4	0.73±0.4	0.6±0.42	0.9±0.26	0±0	0.68±0.36	0.95±0.16	0.76±0.4	0.66±0.39
	All	0.52±0.43	0.52±0.43	0.51±0.42	0.45±0.43	0.63±0.41	0±0	0.46±0.4	0.72±0.37	0.59±0.42	0.44±0.42
Boss Difficulty	RCG	5.41±1.68	5.28±1.59	5.75±1.91	5.39±1.73	5.44±1.63	2.8±1.48	5.86±1.42	5.61±1.38	5.48±1.31	5.33±2.03
	PCG	3.05±2.09	2.84±2	3.58±2.31	3.61±2.25	2.06±1.34	6.2±1.79	3.43±1.96	1.72±0.9	2.96±2.16	3.14±2.06
	All	4.23±2.23	4.06±2.17	4.67±2.35	4.5±2.18	3.75±2.26	4.5±2.37	4.64±2.09	3.67±2.28	4.22±2.18	4.24±2.3
Design	RCG	4.72±1.66	4.53±1.64	5.22±1.66	4.63±1.79	4.88±1.42	4.6±2.23	4.73±1.7	4.74±1.54	4.17±1.61	5.32±1.53
	PCG	3.53±1.47	3.54±1.48	3.5±1.5	3.67±1.45	3.29±1.51	3.27±1.46	3.57±1.4	3.56±1.62	3.3±1.47	3.78±1.45
	All	4.13±1.67	4.04±1.63	4.36±1.78	4.15±1.69	4.08±1.65	3.93±1.91	4.15±1.64	4.15±1.67	3.74±1.59	4.55±1.67
Fun	RCG	4.35±1.99	4.13±2.05	4.96±1.76	4.18±1.98	4.66±2.03	4.2±2.17	4.29±1.96	4.47±2.09	4.09±1.92	4.64±2.07
	PCG	3.4±1.81	3.38±1.89	3.46±1.67	3.39±1.73	3.41±2.01	2.1±1.34	3.57±1.65	3.56±2.04	3.04±1.8	3.79±1.79
	All	3.88±1.95	3.75±1.99	4.21±1.85	3.79±1.89	4.03±2.09	3.15±2.03	3.93±1.82	4.01±2.09	3.57±1.91	4.21±1.96
Inmersibility	RCG	4.35±1.98	4.09±2.16	5.04±1.23	4.11±1.96	4.78±2.01	3.6±1.98	4.43±1.75	4.47±2.28	4.17±1.84	4.55±2.16
	PCG	4.16±1.81	4.06±1.78	4.42±1.94	4.16±1.66	4.16±2.1	3.4±2.27	4.38±1.58	4.11±1.97	4.07±1.71	4.26±1.94
	All	4.26±1.89	4.08±1.96	4.73±1.62	4.13±1.8	4.47±2.04	3.5±2.01	4.41±1.65	4.29±2.11	4.12±1.76	4.41±2.03
Comment Length	RCG	200.5±274.97	120.09±136.41	414.92±417.33	204.86±320.86	192.88±177.02	121.2±163.77	201.67±351.68	221.17±192.69	236.48±345.66	161.1±167.37
	PCG	177.22±222.65	85.66±80.27	336±155.91	159.57±155.57	144.06±154.53	123.4±170.03	176.67±169.8	135.89±132.94	148±171.54	160.43±135.09
	All	200.5±274.97	102.88±112.37	375.46±310.72	182.21±250.89	168.47±165.32	122.3±157.39	189.17±273.05	178.53±168.78	192.24±273.5	160.76±150.23
Comment Type	RCG	2.68±1.55	2.41±1.6	3.42±1.17	2.64±1.59	2.75±1.53	1.6±1.82	2.38±1.6	3.33±1.19	2.61±1.62	2.76±1.51
	PCG	2.55±1.62	1.94±1.63	3.67±1.16	2.32±1.7	2.56±1.71	1.6±2.19	2.38±1.75	2.67±1.5	2.09±1.62	2.76±1.73
	All	2.68±1.55	2.17±1.62	3.54±1.14	2.48±1.64	2.66±1.6	1.6±1.9	2.38±1.65	3±1.37	2.35±1.62	2.76±1.61

(Players and Developers); for **Developing games** with two alternatives: subjects who perform video game development tasks less than 10 h per week (<10h/week) and subjects who dedicate more than 10 hours per week to these activities (>10h/week); **Gamer Profile**, with three alternatives: subjects with a player profile close to the target public of the game in which the evaluated bosses are contextualized (3), subjects with a player profile neutral (2) and subjects with a profile far removed from the target audience (1); and **Group**, whose two alternatives reflect the sequence in which subjects have played and evaluated the bosses generated with each technique (G1: RCG-PCG, G2: PCG-RCG). Note that Table 3 also shows the values of means and standard deviations by combination of the factor **Technique** with the confounding factors. This allows us to illustrate both the effects that the confounding factors have on the evaluation of a boss and the effects that they can have on the evaluation of the differences of bosses performed with different techniques.

To quantify the differences in the response variables due to each factor, we analyzed the Cohen d values. Table 4 shows the Cohen d values of the response variables for all of the fixed factors considered in the statistical analysis. Positive values indicate differences in favor of the first alternative of the factors and negative values indicate differences in favor of the second alternative of the factor. Values indicating a small, medium or large effect due to a factor are highlighted in light, medium and dark gray, respectively. In the case of the factor **Gamer Profile**, with three alternatives, the table shows the Cohen d values of all two-to-two comparisons of these alternatives. The values are shown in an order triad, where the Cohen

Table 4: Cohen d values for the response variables for each fixed factor. Gamer Profile: 1=Non Target audience, 2=Neutral, and 3=Target audience

	Technique (RCG vs PCG)	**Profile** (Players vs Developers)	**Developing Games** ($< 10hweek$ vs $\geq 10hweek$)	**Gamer Profile** (1vs2, 1vs3, 2vs3)	**Group** (G1 vs G2)
Game duration	0.941	0.086	0.103	(0.203,-0.213,-0.448)	0.051
Won rate	-1.024	0.010	-0.434	(-1.265,-2.166,-0.667)	0.353
Boss difficulty	1.248	-0.272	0.339	(-0.067,0.363,0.448)	-0.009
Design	0.760	-0.194	0.039	(-0.128,-0.125,0.002)	-0.497
Fun	0.501	-0.235	-0.125	(-0.418,-0.417,-0.044)	-0.335
Immersibility	0.102	-0.347	-0.177	(-0.527,-0.379,0.060)	-0.151
Comment Length	0.209	-1.456	0.061	(-0.261,0.338,0.046)	0.141
Comment Type	0.168	-0.910	-0.541	(-0.460,-0.936,-0.405)	-0.257

d values between alternatives 1 and 2, 1 and 3, and 2 and 3 of the factor are shown in this order.

According to the Cohen d values of the response variables for **Technique** (first column of Table 4), we can affirm that the effect size of this factor for *Game Duration*, *Won rate*, and *Boss Difficulty* was large, with Cohen d values of 0.941, -1.024 and 1.248, respectively. The signs of these values indicate that the subjects' *Game duration* were longer with the RCG boss than with the PCG boss, but that the *Won rate* is significantly lower, they win less often because the *Boss difficulty* of the RCG boss is higher than that of the PCG boss. The effect size of the factor **Technique** in favor of the RCG boss was medium for *Design* and *Fun* and negligible for the rest of variables with with Cohen d values of less or around 2.

Table 4 also shows the Cohen d values of the response variables for the confounding factors considered in the statistical analysis. The first six rows of the table show how the confounding factors has no effects on all the response variables related to the quality perceived by subjects and that these effects are only large in the case of **Gamer Profile** for *Won rate*. The forth column of Table 4 shows that the factor **Gamer Profile** have effects in all the response variables except in *Design*. Cohen d values of *Won rate*, *Fun* or *Immersibility* indicate that subjects with a profile farther away to the target audience (Alternative 1 of the factor) have a much lower *Won rate* than subjects closer from the target audience, in fact they didn't actually win any games (see the sixth column of the second row of Table 3). Subjects with non target audience profile also score worse on *Fun* or *Immersibility* variables. In *Fun* and *Immersibility* the differences between factor alternatives 2 and 3, neutral subjects or subjects closer to the target audience respectively, are negligible.

The values of the second column of Table 4 shown that the factor **Profile** has large effects on *Comment length* and *Comment type* in favor of developers. Developers made longer and better quality comments than players. The Cohen d values of the last two rows of the table, corresponding to the variables related to the quality of the subjects' comments, indicate that the best comments also come from subjects who spend more time **developing games** and from subjects with a **gamer profile** that is closer to the target audience.

5.2 Hypothesis Testing and Response to the Research Questions

The statistical linear mixed models used to explain the statistical significance of the changes in the response variables are different for each one of them. We selected the statistical models that obtained higher values for the AIC and BIC fit statistics from among all those that do verify the normality of the residuals. In addition, the use of the Linear Mixed Model test assumed that residuals must be normally distributed. All of the residuals, except the ones carried out for *Game duration* and *Comment length*, obtained a p-value greater than 0.05 with the normality test. We obtained normally distributed residuals for *Game duration* and *Comment length* by using neperian logarithm transformation and cubic root transformation respectively. For the statistical analysis of this variables with LMM, we used $RV = \ln(Comment\ length)$ and $RV = \sqrt[3]{Commentlength}$ in formula (1). For the rest of the variables, RV is equal to their value.

Table 5 shows the results of the Type III fixed effects test for each of the response variables or transformations, and for each fixed factor of the statistical model used in each case. Factors or combinations of factors that are not present in the statistical model used to explain the variable are marked with the value NA. Combinations of factors that were not part of the statistical models used are not included in the table. Values indicating significant differences are shaded in grey. According to the results show in Table 5, not all the factors included in the statistical models that explain the response variables produce significant changes in them. For example, to explain the variable *Game duration*, the statistical model used on the transformation of the variable ($DV = \sqrt[3]{Commentlength}$) was $RV \sim Tech.{+}DevGames{+}GamerP{+}Tech.*DevGames{+}\left(1|Subj.\right)$ with the fixed factors **Technique**, **Developing Games**, and **Gamer Profile**, and the combination of factor **Technique** and **Developing Games**, but there are significant differences in the response variable only for the factor **Technique** and the combination **Technique** and **Developing Games**. The changes in the *Game duration* due to the **Technique** used to create the boss being played are statistically significant, just as there are significant differences between the differences between the time spent playing each boss (RCT or PCT) as a function of the time spent developing video games (the alternatives of **Developing games**. As shown by the means and standard deviations of the time spent playing each boss as a function of the time spent developing video games (see first three rows of third column of Table 3). Subjects who spend less time developing software played more time with the RCG boss and less time with the PCG boss than the time that subjects who spend more time developing video games spent playing with the same bosses.

For all the response variables related to the quality perceived by subjects, except for *Immersibility*, the differences due to **Technique** were statistically significant with p values of less than 0.05. Therefore, we can answer our first research question ***RQ1*** rejecting our first null hypothesis, $H_{0,1}$. The two techniques compared in the experiment, RCG and PCG, result in bosses with different quality perceived by the subjects, and it can be concluded that the **Technique** have effects on the perceived *Quality* of the game. The effect size and direction of these differences described in the previous subsection, suggest that the subjects perceive the boss generated by RCG to be of superior quality in comparison to the one generated with PCG.

Whith regard to the second research question, ***RQ2***, the answer is that the null hypothesis $H_{0,2}$ cannot be completely reject. Our results cannot confirm that the **Evaluator's profile**, represented by **Profile**, **Developing Games**, and **Gamer Profile**, has significant effect on the evaluation of the *Quality* of a game. The results indicated that no significant changes were observed in the majority of the response variables used to evaluate the quality of bosses. The only statistically significant changes were observed in the comments made by the subjects and in the won rate.

Not all of the confounding factors considered in the statistical analysis cause statistically significant differences in the response variables. In fact, for the factors related to the Evaluators profile, **Profile**, **Developing Games**, and **Gamer Profile**, no statistically significant differences were confirmed in any of the variables related to the quality perceived by subjects, with the exception of *Won rate* and *Game duration*. The p-value of less than 0.001 for the factor **Gamer Profile** in *Won rate* confirms the statistical significance that could be inferred in the previous subsection from the large effect size of the differences in the variable due to this factor. Subjects who were the furthest from the target audience of the game did not win their games, while the closer the Gamer profile was to the target audience, the more the won rate increased. However, there were not significant differences due to **Gamer Profile**, nor due to **Profile** or **Developing games**, in the evaluation of *Boss difficulty*, *Design*, *Fun*, or *Immersibility*.

However, there are statistically significant changes in the variables related to the subjects' comments due to the factors **Profile** and **Gamer Profile**. The p values of less than 0.05 for *Comment*

Table 5: Results of the Type III test of fixed effects for each response variable and factor or factor's interactions. NA=Not Applicable

	Technique (Tech.)	**Profile**	**Developing Games (DevGames)**	**Gamer Profile (GamerP)**	**Group**	**Tech.*Profile**	**Tech.*DevGames**	**Tech.*GamerP**	**Tech.*Group**
ln(*Game Duration*)	F=43.369 ; p=<.001	NA	0.818;p=0.371	F=1.44; p=0.25	NA	NA	F=6.585; p=0.014	NA	NA
Won rate	F=38.542 ; p=<.001	F=1.884; p=0.178	NA	F=26.034; p=<.001	F=3.322; p=0.076	NA	NA	NA	NA
Boss Difficulty	F=30.358; p=<.001	F=1.299; p=0.261	NA	F=2.281; p=0.116	F=0.203; p=0.655	NA	NA	NA	NA
Design	F=16.445; p=<.001	F=0.257; p=0.615	F=0.575; p=0.453	F=0.081;p=0.922	F=4.301 ; p=0.045	NA	NA	NA	NA
Fun	F=8.199; p=0.007	NA	NA	F=0.666;p=0.519	NA	NA	NA	F=0.696; p=0.504	NA
Inmersibility	F=0.702; p=0.407	F=1.064;p=0.309	F=0.004; p=0.952	F=0.534;p=0.59	F=0.145; p=0.706	NA	NA	NA	NA
$\sqrt[3]{CommentLength}$	F=2.108 ; p= 0.154	F=27.315; p=<.001	F=2.104 ;p=0.155	F=3.784 ; p=0.031	NA	NA	NA	NA	NA
Comment Type	F=1.455; p= 0.234	F=18.069;p=<.001	F=3.564 ;p=0.067	F=7.959;p=0.001	F=2.692; p=0.109	NA	NA	NA	NA

length and *Comment type* in the last two rows of the second and fourth columns of Table 5 confirm the statistical significance of these differences. Developers and subjects with a gamer profile that is closer to the target audience made statistically significant longer and better quality comments than players or, in particular, subjects further away from the game's target audience.

6 DISCUSSION

In the context of video games, reuse is not perceived as a completely positive practice. In fact, developers fear that reusing might be perceived as repetitive by players. On the other hand, the stochastic nature of PCG is perceived positively as an extension in the range of the creativity space for new content. Our experiment shows that this negative view of reuse is not aligned with the results. On the contrary, it reinforces the RCG pathway which boosts the latent content and leads to better results than PCG. During the focus group, subjects agree on that RCG was a natural evolution of the original content. In contrast, PCG was negatively classified as content that did not appear to have been developed by professional developers.

Previous studies considered only players as the subjects of the experiments. In our experiment, we go one step ahead and analyse the differences between players and developers. For researchers, it can be difficult to find developers to run experiments. However, that could not be the case for development studios. For instance, a large studio can enroll developers from different projects from the studio. This is relevant for studios because they put a lot of effort into enrolling players (not developers) for their games. It may seem paradoxical that it is hard to find players, but the experience of testing parts of a game in development is not the same as testing a full game as the developers in the focus group pointed out. Our experiment reveals that there are no relevant differences in terms of statistical values between players and developers, suggesting that studios can leverage their developers. Furthermore, when it comes to feedback developers provided more beneficial feedback as the focus group acknowledge.

This experiment combines the specific quality aspects of video games ('design', 'difficulty', 'fun', and 'immersibility') and the rigorousness of more traditional software work. This includes the replication package that we have not found in previous work. One may think that the complexity of video games makes it difficult to design packages for replication. Nevertheless, we expect that our work along with the replication package available will provide a basis and inspiration for future researchers of the game software engineering community.

7 THREATS TO VALIDITY

To describe the threats to validity of our work, we use the classification of Wohlin *et al.* [58]. This section shows the threats that affected the experiment.

Conclusion validity: The *low statistical power* was minimized because the confidence interval is 95%. The *reliability of measures* threat was mitigated because the measurements were obtained from the data sheets that were automatically generated by the forms with the answers of the subjects when they performed the tasks.

The *reliability of treatment implementation* threat was alleviated because the procedure was identical in all the sessions of the experiment.

Internal validity: To avoid the *instrumentation* threat, we conducted a pilot study to verify the design and the instrumentation. The *interactions with selection* threat affected the experiment because there were subjects who had different levels of experience and, in general, different levels of knowledge of the video game domain. To mitigate this threat, the treatments were applied randomly and the statistical analysis includes the analysis of confounding factors related to subjects' profile.The effects of the design factors, sequence and period, also have been included in the statistical analysis though the analysis of the factors **Group** (Sequence) and **Technique*Group** (Period). Only the variable *Design* has significant changes due to the factor **Group**. This threat also affected the experiment because of the voluntary nature of participation. We selected students from a course whose content was in line with the experiment activities to avoid student demotivation.

Construct validity: To mitigate the *mono-method bias* threat, we mechanized the measurements as much as possible by means of correction templates. To weaken the *evaluation apprehension* threat, at the beginning of the experiment, the instructor explained to the subjects that the experiment was not a test of their abilities. The instructor also told the students that neither participation nor results would affect their grades in the course where the experiment took place. In order to mitigate the *author bias* threat, the tasks were extracted from a commercial video game and were designed by the same experts with similar difficulty for the two methods compared. The experiment was affected by the *mono-operation bias* threat since we only compare two representative bosses of each content generation technique.

External validity: The *interaction of selection and treatment* threat affects the experiment because it involves a different number of subjects in each alternative of the confounding factors. The players are more represented in the overall results than developers. The *domain* threat occurs because the experiment has been conducted in a specific domain (video game) and for a very specific type of game, a spacial shooter, Kromaia. We think that other experiments in different games should be performed to validate our findings, and we hope that this experiment and its replication package will help to perform them.

8 CONCLUSION

In this work, we present an experiment that compares PCG and RCG in terms of design, difficulty, fun and immersibility. This work investigates the influence of PCG and RCG on content generation, employing the video game Kromaia (a commercial video game). Our study suggests that the comparison between RCG and PCG techniques yielded bosses with differing quality levels according to the subjects' perceptions. Specifically, the results demonstrate that the boss produced by RCG is viewed as superior in quality when compared to the one generated by PCG. Moreover, there is no statistical distinction in boss quality based on subject profile (players and developers). However, significant statistical variances are observed in the comments given by the subjects and in the win rate against the boss.

Availability Replication package is at:

Acknowledgements Omitted for blind review.

REFERENCES

[1] Diaz-Furlong Hector Adrian and Solis-Gonzalez Cosio Ana Luisa. 2013. An approach to level design using procedural content generation and difficulty curves. In *2013 IEEE Conference on Computational Inteligence in Games (CIG)*. IEEE, 1–8.

[2] Apostolos Ampatzoglou and Ioannis Stamelos. 2010. Software engineering research for computer games: A systematic review. *Information and Software Technology* 52, 9 (2010), 888–901.

[3] Apostolos Ampatzoglou and Ioannis Stamelos. 2010. Software engineering research for computer games: A systematic review. *Information and Software Technology* 52, 9 (2010), 888–901.

[4] Earl T Barr, Mark Harman, Yue Jia, Alexandru Marginean, and Justyna Petke. 2015. Automated software transplantation. In *Proceedings of the 2015 International Symposium on Software Testing and Analysis*. 257–269.

[5] Nicolas A. Barriga. 2019. A Short Introduction to Procedural Content Generation Algorithms for Videogames. *International Journal on Artificial Intelligence Tools* 28, 2 (2019), 1–11. https://doi.org/10.1142/S0218213019300011

[6] Victor R. Basili and H. Dieter Rombach. 1988. The TAME Project: Towards Improvement-Oriented Software Environments. *IEEE Transactions on Software Engineering* (1988).

[7] Daniel Blasco, Jaime Font, Mar Zamorano, and Carlos Cetina. 2021. An evolutionary approach for generating software models: The case of Kromaia in Game Software Engineering. *Journal of Systems and Software* 171 (2021), 110804.

[8] Unreal Blueprint. [n. d.]. Unreal Blueprint. https://docs.unrealengine.com/4.27/en-US/ProgrammingAndScripting/Blueprints/GettingStarted/. Accessed: 01/02/24.

[9] Joseph Alexander Brown and Marco Scirea. 2022. Evolving Woodland Camouflage. *IEEE Transactions on Games* (2022).

[10] Cameron Bolitho Browne. 2008. *Automatic generation and evaluation of recombination games*. Ph. D. Dissertation. Queensland University of Technology.

[11] Luigi Cardamone, Daniele Loiacono, and Pier Luca Lanzi. 2011. Interactive evolution for the procedural generation of tracks in a high-end racing game. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*. 395–402.

[12] Megan Charity, Ahmed Khalifa, and Julian Togelius. 2020. Baba is y'all: Collaborative mixed-initiative level design. In *2020 IEEE Conference on Games (CoG)*. IEEE, 542–549.

[13] Jorge Chueca, Javier Verón, Jaime Font, Francisca Pérez, and Carlos Cetina. 2023. The consolidation of game software engineering: A systematic literature review of software engineering for industry-scale computer games. *Information and Software Technology* (2023), 107330.

[14] Jacob Cohen. 1988. Statistical power for the social sciences. *Hillsdale, NJ: Laurence Erlbaum and Associates* (1988).

[15] CryEngine. [n. d.]. CryEngine. https://www.cryengine.com. Accessed: 01/02/24.

[16] Steve Dahlskog and Julian Togelius. 2013. Patterns as objectives for level generation. In *Design Patterns in Games (DPG), Chania, Crete, Greece (2013)*. ACM Digital Library.

[17] Edirlei Soares de Lima, Bruno Feijó, and Antonio L Furtado. 2019. Procedural Generation of Quests for Games Using Genetic Algorithms and Automated Planning.. In *SBGames*. 144–153.

[18] Edirlei Soares de Lima, Bruno Feijó, and Antonio L Furtado. 2022. Procedural generation of branching quests for games. *Entertainment Computing* 43 (2022), 100491.

[19] África Domingo, Jorge Echeverría, Óscar Pastor, and Carlos Cetina. 2021. Comparing UML-Based and DSL-Based Modeling from Subjective and Objective Perspectives. In *Advanced Information Systems Engineering*. Springer, 483–498.

[20] Lucas Nascimento Ferreira and Claudio Fabiano Motta Toledo. 2017. Tanager: A generator of feasible and engaging levels for Angry Birds. *IEEE Transactions on Games* 10, 3 (2017), 304–316.

[21] Stefan Fischer, Lukas Linsbauer, Roberto E Lopez-Herrejon, and Alexander Egyed. 2015. The ECCO tool: Extraction and composition for clone-and-own. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, Vol. 2. IEEE, 665–668.

[22] Miguel Frade, Francisco Fernandéz de Vega, Carlos Cotta, et al. 2009. Breeding terrains with genetic terrain programming: the evolution of terrain generators. *International Journal of Computer Games Technology* 2009 (2009).

[23] Roberto Gallotta, Kai Arulkumaran, and LB Soros. 2022. Evolving spaceships with a hybrid L-system constrained optimisation evolutionary algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. 711–714.

[24] Daniele Gravina and Daniele Loiacono. 2015. Procedural weapons generation for Unreal Tournament III. In *2015 IEEE Games entertainment media conference (GEM)*. IEEE, 1–8.

[25] J. Hair, R. Anderson, B. Black, and B. Babin. 2016. *Multivariate Data Analysis*. Pearson Education. https://books.google.es/books?id=LKOSAgAAQBAJ

[26] Mark Hendrikx, Sebastiaan Meijer, Joeri Van Der Velden, and Alexandru Iosup. 2013. Procedural content generation for games: A survey. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 9, 1 (2013), 1–22.

[27] Charlene Jennett, Anna L Cox, Paul Cairns, Samira Dhoparee, Andrew Epps, Tim Tijs, and Alison Walton. 2008. Measuring and defining the experience of immersion in games. *International journal of human-computer studies* 66, 9 (2008), 641–661.

[28] Misaki Kaidan, Chun Yin Chu, Tomohiro Harada, and Ruck Thawonmas. 2015. Procedural generation of angry birds levels that adapt to the player's skills using genetic algorithm. In *2015 IEEE 4th Global Conference on Consumer Electronics (GCCE)*. IEEE, 535–536.

[29] Evrim Itir Karac, Burak Turhan, and Natalia Juristo. 2019. A Controlled Experiment with Novice Developers on the Impact of Task Description Granularity on Software Quality in Test-Driven Development. *IEEE Transactions on Software Engineering* (2019).

[30] Vid Kraner, Iztok Fister Jr, and Lucija Brezočnik. 2021. Procedural content generation of custom tower defense game using genetic algorithms. In *International Conference "New Technologies, Development and Applications"*. Springer, 493–503.

[31] Jialin Liu, Sam Snodgrass, Ahmed Khalifa, Sebastian Risi, Georgios N Yannakakis, and Julian Togelius. 2021. Deep learning for procedural content generation. *Neural Computing and Applications* 33, 1 (2021), 19–37.

[32] Carlos López-Rodríguez, Antonio J Fernández-Leiva, Raúl Lara-Cabrera, Antonio M Mora, and Pablo García-Sánchez. 2020. Checking the Difficulty of Evolutionary-Generated Maps in a N-Body Inspired Mobile Game. In *International Conference on Optimization and Learning*. Springer, 206–215.

[33] Tuhin Das Mahapatra. 2023. Why is Rockstar delaying GTA 6? Here are some possible breakdowns. https://www.hindustantimes.com/technology/why-is-rockstar-delaying-gta-6-here-are-some-possible-breakdowns-101681440818791.html. Accessed: 01/02/24.

[34] Carlos Mora, Sandra Jardim, and Jorge Valente. 2021. Flora Generation and Evolution Algorithm for Virtual Environments. In *2021 16th Iberian Conference on Information Systems and Technologies (CISTI)*. IEEE, 1–6.

[35] Peter Naur and Brian Randell. 1969. *Software Engineering: Report of a conference sponsored by the NATO Science Committee, Garmisch, Germany, 7-11 Oct. 1968, Brussels, Scientific Affairs Division, NATO*.

[36] Peter Thorup Ølsted, Benjamin Ma, and Sebastian Risi. 2015. Interactive evolution of levels for a competitive multiplayer FPS. In *2015 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 1527–1534.

[37] Leonardo Tortoro Pereira, Paulo Victor de Souza Prado, Rafael Miranda Lopes, and Claudio Fabiano Motta Toledo. 2021. Procedural generation of dungeons' maps and locked-door missions through an evolutionary algorithm validated with players. *Expert Systems with Applications* 180 (2021), 115009.

[38] Leonardo T Pereira, Breno MF Viana, and Claudio FM Toledo. 2021. Procedural enemy generation through parallel evolutionary algorithm. In *2021 20th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)*. IEEE, 126–135.

[39] David Plans and Davide Morelli. 2012. Experience-driven procedural music generation for games. *IEEE Transactions on Computational Intelligence and AI in Games* 4, 3 (2012), 192–198.

[40] Klaus Pohl and Andreas Metzger. 2018. Software product lines. *The Essence of Software Engineering* (2018), 185–201.

[41] Cristiano Politowski, Fabio Petrillo, João Eduardo Montandon, Marco Tulio Valente, and Yann-Gaël Guéhéneuc. 2021. Are game engines software frameworks? A three-perspective study. *Journal of Systems and Software* 171 (2021), 110846.

[42] Hafizh Adi Prasetya and Nur Ulfa Maulidevi. 2016. Search-based Procedural Content Generation for Race Tracks in Video Games. *International Journal on Electrical Engineering & Informatics* 8, 4 (2016).

[43] Piotr Rykała. 2020. The growth of the gaming industry in the context of creative industries. *Biblioteka Regionalisty* 20 (2020), 124–136.

[44] Ronnie ES Santos, Cleyton VC Magalhães, Luiz Fernando Capretz, Jorge S Correia-Neto, Fabio QB da Silva, and Abdelrahman Saher. 2018. Computer games are serious business and so is their quality: particularities of software testing in game development from the perspective of practitioners. In *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. 1–10.

[45] Unity Scripting. [n. d.]. Unity Scripting. https://unity.com/features/unity-visual-scripting. Accessed: 01/02/24.

[46] Bran Selic. 2003. The pragmatics of model-driven development. *IEEE software* 20, 5 (2003), 19–25.

[47] Adam Summerville, Sam Snodgrass, Matthew Guzdial, Christoffer Holmgard, Amy K. Hoover, Aaron Isaksen, Andy Nealen, and Julian Togelius. 2018. Procedural Content Generation via Machine Learning (PCGML). *IEEE Transactions on Games* 10, 3 (2018), 257–270. https://doi.org/10.1109/tg.2018.2846639 arXiv:1702.00539

[48] Barbara G Tabachnick, Linda S Fidell, and Jodie B Ullman. 2007. *Using multivariate statistics*. Vol. 5. Pearson Boston, MA.

[49] Walter F Tichy. 1998. Should computer scientists experiment more? *Computer* 31, 5 (1998), 32–40.

[50] Julian Togelius, Mike Preuss, Nicola Beume, Simon Wessing, Johan Hagelbäck, Georgios N Yannakakis, and Corrado Grappiolo. 2013. Controllable procedural map generation via multiobjective evolution. *Genetic Programming and Evolvable Machines* 14, 2 (2013), 245–277.

[51] Julian Togelius, Georgios N Yannakakis, Kenneth O Stanley, and Cameron Browne. 2011. Search-based procedural content generation: A taxonomy and survey. *IEEE Transactions on Computational Intelligence and AI in Games* 3, 3 (2011), 172–186.

[52] Julian Togelius, Georgios N. Yannakakis, Kenneth O. Stanley, and Cameron Browne. 2011. Search-based procedural content generation: A taxonomy and survey. *IEEE Trans. on Computational Intelligence and AI in Games* 3, 3 (2011), 172–186.

[53] Sira Vegas, Cecilia Apa, and Natalia Juristo. 2015. Crossover designs in software engineering experiments: Benefits and perils. *IEEE Transactions on Soft. Eng.* 42, 2 (2015), 120–135.

[54] Breno MF Viana and Selan R dos Santos. 2019. A survey of procedural dungeon generation. In *2019 18th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)*. IEEE, 29–38.

[55] Breno MF Viana, Leonardo T Pereira, and Claudio FM Toledo. 2022. Illuminating the space of enemies through map-elites. In *2022 IEEE Conference on Games (CoG)*. IEEE, 17–24.

[56] Steve Watts. 2020. All The Cyberpunk 2077 Delays. https://www.gamespot.com/gallery/all-the-cyberpunk-2077-delays/2900-3618/. Accessed: 01/02/24.

[57] Brady T West, Kathleen B Welch, and Andrzej T Galecki. 2014. *Linear mixed models: a practical guide using statistical software*. Chapman and Hall/CRC.

[58] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. 2012. *Experimentation in software engineering*. Springer Science & Business Media.

[59] Georgios N Yannakakis and Julian Togelius. 2018. *Artificial intelligence and games*. Vol. 2. Springer.

[60] Meng Zhu and Alf Inge Wang. 2019. Model-driven game development: A literature review. *ACM Computing Surveys (CSUR)* 52, 6 (2019), 1–32.