Game Software Engineering: A Controlled Experiment Comparing Automated Content Generation Techniques

Mar Zamorano
maria.lopez.20@ucl.ac.uk
University College London
London, UK

África Domingo
adomingo@usj.es
Universidad San Jorge
Zaragoza, Spain

Carlos Cetina
cetina@upv.es
Universitat Politècnica de València
Valencia, Spain
University College London
London, UK

Federica Sarro
f.sarro@ucl.ac.uk
University College London
London, UK

ABSTRACT

Background Video games are complex projects that involve a seamless integration of art and software during the development process to compose 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 'immersiveness', which are not considered for traditional software. On the other hand, there are not well-established best practices for the empirical assessment of video games as 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 video games following best practices established in software engineering research. 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 generated content. **Method** We conducted a controlled experiment where human subjects had to play with content that was automatically generated for a commercial video game by the two techniques (PCG and RCG), and evaluate it according to specific quality aspects of video games. A total of 44 subjects including professional developers and players participated in our experiment. **Results** The results suggest that participants perceive that RCG generates content is of higher quality than PCG. **Conclusions** The results can turn the tide for content generation. So far, RCG has been neglected as a viable option: typically, reuse is frowned upon by the developers, who aim to avoid repetition in their video games as much as possible. However, our study uncovered that RCG unlocks latent content that is actually favoured by players and developers alike. This revelation poses an opportunity towards opening new horizons for content generation research.

ESEM '24, October 24–25, 2024, Barcelona, Spain

ACM ISBN 979-8-4007-1047-6/24/10.
https://doi.org/10.1145/3674805.3686690

KEYWORDS

Empirical Study, Game Software Engineering, Video Game

ACM Reference Format:
Mar Zamorano, África Domingo, Carlos Cetina, and Federica Sarro. 2024. Game Software Engineering: A Controlled Experiment Comparing Automated Content Generation Techniques. In *Proceedings of the 18th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM '24), October 24–25, 2024, Barcelona, Spain.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3674805.3686690

1 INTRODUCTION

The video games industry is in continuous growth every year [42]. Despite being one of the fastest growing industries, game software engineering (GSE) has been identified as an area that needs more fundamental research [2, 11]. One of the aspects where GSE needs more rigorous research pertains empirical research methods [11].

While theoretical frameworks provide a foundational understanding, empirical studies offer the necessary validation and refinement, which is crucial for effective implementations. 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 [46, 57]. By empirically investigating the user experience of video game techniques, researchers can unveil 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 most pressing challenges in video game development is the need for new content [48]. Video game content generation is often a slow, laborious, costly, and error-prone process. This results in issues such as significant delays in content development [31, 55] or the need to introduce game content in post-launch updates. Through rigorous experimentation, empirical studies can serve as the cornerstone for pushing the boundaries of content generation.

In this study, we aim to empirically assess and compare content generated by two different content generation techniques along with two different user profiles (players and developers). We study two automated approaches for generating content, Procedural Content Generation (PCG) and Reuse-based Content Generation (RCG), and whether their use has an impact on the quality of the generated content. We do so by analyzing the commercial video game Kromaia

released on PlayStation 4 and Steam. Specifically, we invite participants (developers and players) to play with content generated by PCG and RCG in the game, and then evaluate their experience in terms of video game specific percieved quality measures, namely 'difficulty', 'design','fun', and 'immersiveness'[17]. Participants were not told how the content was generated. We conducted 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. A total of 44 participants took part in the experiment, assessing the generated content in two scenarios of the game. The results show that the participants perceive the content generated by RCG to be of superior quality in comparison to the content generated by PCG: RCG obtained better results than PCG in 77% of the cases based on *difficulty*, in 34% cases for *design*, in 28% cases for *fun* and 5% for *immersiveness*.

Our findings challenge three prevailing trends in GSE. Firstly, there is a perception that content reuse leads to repetitive game content, which is typically frowned upon by developers. However, our research indicates that subjects actually prefer content generated through RCG. Secondly, previous content generation experiments have involved only the players, neglecting the input of developers. Our results demonstrate no significant differences between players and developers. This suggests that the input of developers is also relevant for content generation. Furthermore, developers are shown to provide more detailed feedback. Lastly, 73% of previous content generation experiments have missed important factors such as hypotheses formulation, statistical analysis, or the inclusion of a replication package. We have not found any reasons for neglecting the aforementioned practices, and hence, our work encompasses all of the above - including replication, which has been overlooked in 100% of previous studies. We hope that our research will inspire future research in GSE to comply with empirical best practices.

The rest of the paper is structured as follows: Section 2 presents the techniques under study and the context of the experiment. Section 3 outlines the experimental design. Section 4 presents the experiment results, followed by a discussion in Section 5. Section 6 summarizes the threats to the validity. Section 7 reviews the related work. Finally, Section 8 concludes the paper.

2 BACKGROUND

The development process of video games requires a harmonious combination of artistic elements and software integration, resulting in intricate and multifaceted creations. Nowadays, most video games are developed by means of game engines. One can argue that game engines are software frameworks [39]. 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 [50] and Unreal Engine [51], but it is also possible for a studio to make its own specific engine (*e.g.*, CryEngine [13]). Developers can use these engines and traditional coding approaches (C++ on Unreal or C# on Unity) to create video game content.

2.1 Content Generation for Video Games

The process of content generation for video games is typically slow, tedious, expensive, and error-prone. In turn, this leads to many problems for the industrial development of video games, which typically impact the consumers in the form of (1) excessive delays in content creation (with notorious examples in Cyberpunk 2077 [55] or GTA VI [31]) and (2) an ever-increasing demand for game content derived from Downloadable Content (DLCs).

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 [23]. 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 , Machine Learning techniques [29, 44] , and Search-Based techniques [49, 59].

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 [33], its evolution has been tied to the concept of reuse, whether through opportunistic approaches such as clone-and-own [18], or systematic approaches such as software product lines (assembling predefined features) [38] or software transplantation (a feature is transplanted from a donor to a host) [4]. A recent SLR on GSE [11] identifies the relevance of both Procedural Content Generation (PCG) and Reuse-based Content Generation (RCG) for content generation in video games.

We carry out our study by using the commercial video game Kromaia released on PlayStation and Steam, translated into eight languages. In the game, 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 in this study, the above mentioned *boss* is the content generated. To do so, developers generate content through PCG by means of the work of Gallota *et al.* (which combines a Lindenmayer systems [28] with an Evolutionary Algorithm) [20]. This approach 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 the game 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 PCG and RCG and the importance of content generation for video game development, there is no study that directly compares both approaches.

3 EXPERIMENTAL DESIGN

In this section we present the experiment design following Wohlin's guidelines [57] for reporting software engineering experiments.

3.1 Objective

The research objective has been organized using the Goal Question Metric (GQM) template for defining objectives originally presented by Basili and Rombach [6]. Our goal is to **analyze** different techniques for content generation, namely 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 players and developers; **in the context of** content generation for an existing video game.

3.2 Research Questions and Hypotheses

The research questions and null hypotheses are 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 - Do evaluators with different profiles evaluate the quality of the game differently? 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, as this is the first time these RQs are studied and there is no reason to assume that one approach is better than the other.

3.3 Variables

In this study, the factor under investigation is the content generation technique (**Technique**) used to automatically generate content, *i.e.*, 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 human participants. 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 participants playing the generated content. We decomposed the analysis of quality into different dimensions: *difficulty*, *design*, *fun* and *immersiveness*, based on previous work [17].

To evaluate difficulty we used three response variables: *Game duration*, *Won rate* and *Boss difficulty*. *Game duration* is the average time spent by each participant in their games. The value of this variable was calculated by dividing the time each participant 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, calculated by dividing the number of games won by the number of games played against a boss. We measured *Boss difficulty* based on the participant'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 *Immersiveness*. Each of these variables correspond to specific items in the questionnaire. The participants 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 1 shows the specific items of the questionnaire, used for the calculation of each of these response variables.

For the evaluation of each boss in the game, the participants also answered an open-ended question in which they could provide additional comments. 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*, the comment type was 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.

Table 1: 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 the game *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 *Item*6. When the time was up, I was disappointed that I could not continue playing against the boss
Immersiveness	*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

In order to establish the different evaluator profiles among the participants, we conducted different sessions of the experiment with specific groups of participants: potential players 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. The groupings of participants in sessions by participant profile (player or developer) and the participants' responses to the demographic questionnaire were used to define three blocking variables: **Profile**, **Game development**, and **Gamer profile**. The objective was to analyze whether and how the experience in video game development and the profile as a player could influence the evaluation of the quality of the game elements.

The blocking variable **Profile** has two alternatives, player or developer, depending on the previous grouping of participants in sessions by profile. This variable also allows the study of the differences between the sessions held and the demographic profiles of the participants. To define the alternatives for the blocking variable **Game development**, the weekly hours that the participants dedicated to developing software for video games were taken into account. The variable will have two alternatives: 1, for participants 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 blocking variable **Gamer profile** is used to distinguish participants with a player profile that is closer to the target audience of the video game being analyzed from participants with less related profiles, such as casual players or those who are not interested in video games. In order to define the alternatives of **Gamer profile** we considered the scores given by the participants 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)

(3) How would you rate your overall experience with shooter video games?(1, No experience; 2, Little experience; 3, Medium experience; 4, Very experienced; and 5, Expert)

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

3.4 Design

We chose a Two-Treament crossover design with two sequences using two different evaluation tasks: T1, evaluate a boss created using RCG, and T2, evaluate a boss created using PCG. The participants were randomly divided into two groups (G1 and G2). In the first period of the experiment, the participants of G1 perform T1 and the participants of G2 perform T2. In the second period, the participants of G1 perform T2 and the participants of G2 perform T1.

This repeated measure design enhances the experiment's sensitivity, as noted by Vegas *et al.* [52]. Considering the same participant evaluating both alternatives, between-participant differences are controlled, thus improving the experiment's robustness regarding variation among participants. 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). We study the effects of the factors period, sequence, and participant to validate of this experiment.

To verify the experiment design, we conducted a pilot study with two participants. 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 participants in the pilot study did not participate in the experiment.

3.5 Participants

We selected the participants using convenience sampling [57]. A total of 46 participants 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 participants included 12 professionals related with video game development and 34 third year undergraduate students who are taking a course in *Software Quality* from different technology programs at a higher education institution (Universidad San Jorge). In particular, part of those students are specifically studying video games design and development.

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.

3.6 Experimental Objects

In the experiment, the participants evaluate content (bosses created for an existing video game). Participants must defeat 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. The player's spaceship is highlighted in orange (see 1 of Figure 1), while the bosses are in black and green (see 2 of Figure 1). The scenario where the player fights the boss is the grey part, and the white balls are projectiles exchanged between the player's spaceship and the boss. The two bosses shown in Figure 1 (PCG boss and RCG boss) are the two best bosses obtained with PCG and RCG according to the game's development team.

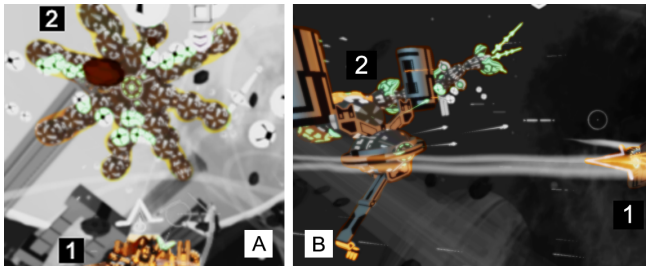

Figure 1: (A) PCG boss. (B) RCG boss.

For the execution of this experiment, a video game engineer who was involved in the development of the game developed a test scenario based on scenarios from the original game. In this scenario, the participants of the experiment can (1) learn how to operate the game controls, (2) learn how to fight an original boss from the game, and (3) fight the 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 sections:
(1) An informed consent form that the participants must review and accept voluntarily. It clearly explains what the experiment consists of and that the personal data will not be collected.
(2) A demographic questionnaire that was used for characterizing the sample and defining the blocking variables.
(3) Specific information on how to download and use the game's test environment that will be used to perform the experiment, and instructions on how to use the game environment.
(4) Specific instructions on how to access the boss fight and the evaluation questionnaire about the game experience against the boss. This section was repeated three times in the questionnaires, once for each boss played by the participants: first against the original boss, and then against the two bosses generated with the techniques we compared (PCG and RCG).

3.7 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 100 minutes and was conducted following the experimental procedure below:
(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 participants' abilities. (5 min)

(2) The participants 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 participants were randomly divided into two groups (G1 and G2). (10 min)
(3) The participants 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 participants completed a demographic questionnaire. (5 min)
(5) The participants received specific information on how to download and use the test environment that will be used to conduct the experiment. They downloaded and used the 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 participants received specific instructions on how to access a fight with an original boss of the game. After playing against the boss, the participants completed the evaluation questionnaire about the experience of playing against the original boss. (15 min)
(7) The participants performed the first task. They received specific instructions on how to access to a fight with the boss to evaluate. The participants of G1 played against the boss generated with RGC while the participants of G2 played against the boss generated with PCG. After playing as many times as desired against the assigned boss, all the participants completed the evaluation questionnaire about the game experience against the boss played. (15 min)
(8) The participants performed the second task. They received instructions on how to access a fight with the boss to evaluate. The participants of G1 played against the boss generated with PCG while the participants of G2 played against the boss generated with RCG. After playing as many times as desired against the assigned boss, all the participants 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.

3.8 Analysis Procedure

We have chosen the Linear Mixed Model (LMM) [56] for the statistical data analysis. LMM handles correlated data resulting from repeated measures, and it allows us to study the effects of factors that intervene in a crossover design (period, sequence, or participant) and the effects of other blocking variables (*e.g.*, in our experiment, profile, game development practice, and gamer profile) [52]. In the hypotheses testing, we applied the Type III test of fixed effects with unstructured repeated covariance.

In this study, **Technique** was defined as a fixed-repeated factor to identify the differences between using PCG or RCG, and the participants were defined as a random factor $(1|Subj)$ to reflect the repeated measures design. The response variables (RV) for this test were as follows: *Game duration*, *Won rate*, *Boss difficulty*, *Design*, *Fun*, and *Immersiveness*, which were related to participants' perceived quality of the boss; *Comment length* and *Comment type*, which were used to determine differences in participants' comments.

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 factor with two alternatives: G1 and G2, corresponding to the two different sequences in which the bosses are evaluated. The first group of participants (G1) played and evaluated the boss generated with RGC, and then played and evaluated the boss generated with PCG. The second group of participants (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 blocking variables related to the evaluators' profile to determine the variability in the response variables, in the statistical model we also considered as fixed factors the blocking variables **Profile**, **Game development**, and **Gamer profile** and the combination of this variables with the principal factor **Technique**.

We tested different statistical models in order to find out which factors or blocking variables, 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 additional fixed factors (AF) considered in the experiment (**Group**, **Profile**, **Game development**, or **Gamer profile**) or their interactions with the factor **Technique** ($Tech.*AF$) which could have effects on the response variables.

$$\begin{array}{ll} Model\,0 & RV \sim Tech.+(1|Subj.) \\ Model\,1 & RV \sim Tech.+AF+Tech.*AF+(1|Subj.) \\ Model\,2 & RV \sim Tech.+AF_1+AF_2+CF_3+AF_4+(1|Subj.) \\ Model\,3 & RV \sim Tech.+AF_1+AF_2+Tech.*AF_1+(1|Subj.) \end{array} \quad (1)$$

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 [16, 26]. 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 response variables due to the fixed factors considered, we calculated the Cohen *d* value [12], which is the standardized difference between the means of the response 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 histograms and boxplots to describe the results graphically.

To verify that the group of measures associated with each response variable or fixed factor is consistent, we applied Principal Components Analysis (PCA) to the set of measures 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 other, 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 [22, 45]. If the group 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 measures. On the other hand, if the measures 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.

4 RESULTS

Principal Component Analysis (PCA) was applied to the set of measures used to define the different response variables and factors of this work. In this work we applied PCA twice, one to the measures used to define the blocking variables, and another to the measures used to define the response variables. The results of this PCA executions are in the replication package. Each component extracted by PCA is a new random variable that summarizes the information of a subset of variables [22, 45]. In general terms, the extracted PCA components were consistent with the subsets of measures selected to define each variable.

The application of PCA to the measures used to define the response variables produced four components. The first component groups mainly the responses to the questions used to define the *Design* and *Fun*, implying similar results in both variables. The second component groups the response variables related to the comments made by the subjects. The third component groups *Won rate* and *Boss Difficulty*, and the fourth component represents the *Game duration*. The responses to the questions used to define immersiveness were part of all of the previous PCA components, but they did not define clearly a single factor.

The application of PCA to the measures used to define the blocking variables **Profile**, **Developing games**, and **Gamer Profile**, produced two components, one defined mainly by the factor **Profile** and the other grouping the responses to the questions used to define **Gamer Profile**. The variability of the factor **Developing games**, related to video game development time, is represented by the two previous components to similar degrees. This means that the variability it contains is explained by both factors, but not only by one of them. We decided to include the factor separately in the statistical analysis even though this result confirms positive correlations with the other two factors under consideration.

4.1 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 *Immersiveness* were small and there were also no large differences due to the factor **Technique** in the variables related to the subjects' comments. Table 2 shows the values for the mean and standard deviation of all the response variables considered (*Game duration*, *Won rate*, *Fun*, *Boss difficulty*, *Design*, *Fun*, *Immersiveness*, *Comment length*, and *Comment type*) for each one of the **Techniques** compared: PCG and RCG, and for each one of the alternatives of the blocking variables and factors considered as fixed factors in the statistical analysis: **Profile**, with two alternatives (Players and Developers); for **Developing games** with two alternatives: subjects who perform video game development tasks for less than 10h 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 2 also shows the values of means and standard deviations by combination of the factor **Technique** with these variables. This allows us to illustrate both the effects that these variables 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. In Table 2 the pairs of values are shaded according to the effect size of their differences. The darker the shade, the larger the difference in the values of the response variables across the alternatives of the factors and blocking variables considered. Additionally, the italicised text highlights the statistically significant comparisons.

To quantify the differences in the response variables due to each factor or blocking variable, we analyzed the Cohen *d* values. Table 3 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 blocking variable **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 *d* values between alternatives 1 and 2, 1 and 3, and 2 and 3 of the blocking variable are shown in this order.

The effect size of a factor measure through the Cohen *d* value is related to the percentage of non-overlap between the distributions of the response variables for each alternative of the factor. Higher effect size correspond with greater percentages of non-overlap and larger differences. The histograms in Figure 2 illustrate the differences in *Won Rate* (left), *Design* (center),and *Immersiveness* (right) depending on the Technique use to generate the boss evaluated. In the *Won Rate* histogram, the non-overlapping parts are around 39%, which corresponds to a very large effect size and to a Cohen *d* value of more than 1. In the *Design* histogram, the non-overlapping parts are around 30%, which corresponds to a large effect size and to a Cohen *d* value of around 0.8. However, in the *Immersiveness* histogram, the non-overlapping parts are around 5%, which corresponds to a negligible effect size and to a Cohen *d* value around 0.

According to the Cohen *d* values of the response variables for **Technique** (first column of Table 3), 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 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 Cohen *d* values of less or around 0.2. Table 3 also shows the Cohen

Table 2: Mean and standard deviation ($\mu \pm \sigma$) values of the dependent variables for the factor (Technique) in each alternative of the fixed factors. The light, medium and dark gray highlight indicates a small, medium or large effect.

		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 (PCT-PCG)	G2 (PCG-PCT)
Game Duration	PCT	*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	PCT	*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	PCT	*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	PCT	*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	PCT	*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
Immersiveness	PCT	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	PCT	200.5±275	120±136	415±417	205±321	193±177	121±164	202±357	*221±193*	236±346	161±167
	PCG	177±223	86±807	336±156	160±156	144±155	123±170	177±170	136±133	148±172	160±135
	All	201±275	*103±112*	*375±311*	182±251	168±165	*122±157*	*189±273*	*179±169*	192±274	161±150
Comment Type	PCT	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

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

	Technique (PCT vs PCG)	Profile (Players vs Developers)	Developing Games (< 10*hweek* vs ≥ 10*hweek*)	Gamer Profile (1vs2, 1vs3, 2vs3)	Group (G1vsG2)
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
Immersiveness	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 of the response variables for the fixed factors considered in the statistical analysis. The first six rows of the table show how the blocking variables 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 bottom part of Figure 2 shows ten pairs of box plots, arranged in rows and columns, illustrating the differences in *Won Rate*, *Design*, and *Immersiveness* due to some of the fixed factors considered. The first row of pairs of box plots corresponds to all of the subjects, and illustrates the differences in the response variables due to **Technique**. The following rows corresponds to the alternatives of the blocking variables considered in each response variable, and illustrates the differences due to this variable and its combination with **Technique**. The boxplots in the bottom left of Fig. 2 illustrate the large effects of the factors **Technique** and the blocking variable **Gamer Profile** in *Won rate*. The box plots in the bottom right of Fig. 2 illustrate the negligible effects of **Technique** (All subjects), and the medium effects of **Gamer Profile** in *Immersiveness*. The blocks of boxplots by fixed factor, after the first row of boxplots, also show the absence of differences of the blocking variables combined with Technique, since the differences between one boss and the other do not depend on the alternative of the variable considered. For *Won rate*, in all alternatives, the won rate is higher or equal with the PCG boss than with the RCG boss, but for *Design* or *Immersiveness*, RCG boss outperforms PCG boss.

The forth column of Table 3 shows that the blocking variable **Gamer Profile** has effects in all the response variables except in *Design*. Cohen *d* values of *Won rate*, *Fun* or *Immersiveness* indicate that subjects with a profile farther away to the target audience (Alternative 1 of the variable) 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 2). Subjects with non target audience profile also score worse on *Fun* or *Immersiveness* variables. In *Fun* and *Immersiveness* the differences between alternatives 2 and 3, neutral subjects or subjects closer to the target audience respectively, are negligible.

The values of the second column of Table 3 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.

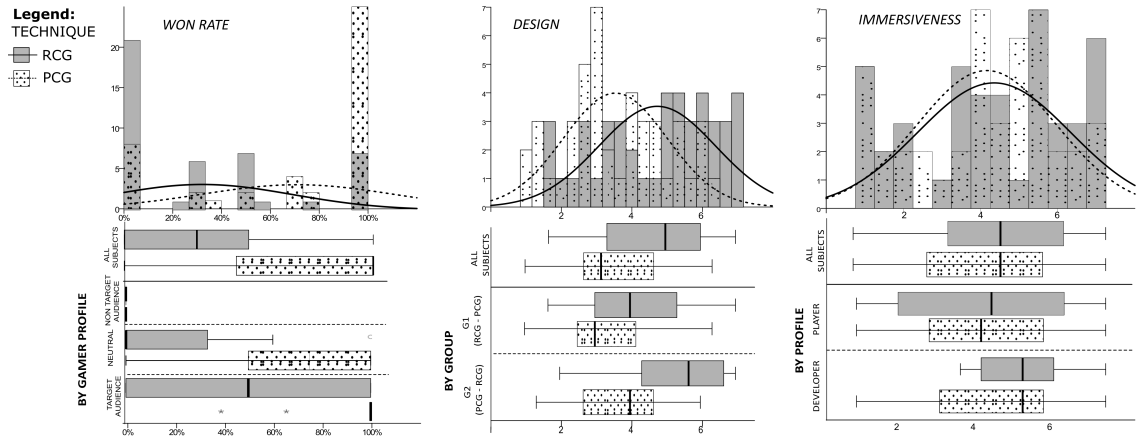

Figure 2: Histograms with normal distributions and box plots for Won Rate, Design and Immersiveness, with boxplots by the alternatives of Gamer Profile, Group and Profile respectively

4.2 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 (LMM) 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 $RV \leftarrow \ln(Comment\ length)$ and $RV \leftarrow \sqrt[3]{Commentlength}$ in formula (1). For the rest of the variables, RV is equal to their value.

Table 4 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 selected to explain the variable are marked with the value NA or are not included in the table. Values indicating significant differences are shaded in grey. According to the results show in Table 4, not all the fixed 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 ($RV \leftarrow \sqrt[3]{Commentlength}$) was $RV \sim Tech.+DevGames+GamerP+Tech.*DevGames+(1|Subj.)$ 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 Table 2 first three rows of third column), 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.

Answer to RQ1. For all the response variables related to the quality perceived by subjects, except for *Immersiveness*, 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** has effects on the perceived *Quality* of the game. The effect size and direction of these differences previously described, suggest that the subjects perceive the boss generated by RCG to be of superior quality in comparison to the one generated with PCG.

Answer to RQ2. With regard to the second research question, ***RQ2***, the answer is that the null hypothesis $H_{0,2}$ cannot be completely rejected. Our results cannot confirm that the **Evaluator's profile**, represented by **Profile**, **Developing Games**, and **Gamer Profile**, has a 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 factors and blocking variables considered in the statistical analysis cause statistically significant differences in the response variables. In fact, for the blocking variables related to the evaluators profile, **Profile**, **Developing Games**, and **Gamer Profile**, no statistically significant differences were confirmed in any of the response 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 **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 response variable due

to **Gamer Profile**. 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 *Immersiveness*.

However, there are statistically significant changes in the variables related to the subjects' comments due to the blocking variables **Profile** and **Gamer Profile**. The p values of less than 0.05 for *Comment length* and *Comment type* in the last two rows of the second and fourth columns of Table 4, 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.

5 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, our results reinforce the RCG path, which boosts the latent content and leads to better results than PCG. During the focus group, subjects agreed 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 beyond 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. 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 acknowledged.

This experiment combines the specific quality aspects of video games ('design', 'difficulty', 'fun', and 'immersiveness') and the rigorousness of more traditional software work. This includes the provisioning of a replication package, something that no previous works did. 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 will serve as a basis and inspiration for future researchers of the GSE community.

6 THREATS TO VALIDITY

We use the classification provided by Wohlin *et al.* [57].

Conclusion Validity: We mitigated possible threats due to *low statistical power* by using a confidence interval of 95% for the statistical analysis. We also mitigated the *reliability of measures* by computing the evaluation measures directly from the data sheets automatically generated from the on-line questionnaire answers provided by the participants. Finally, we use an identical procedure in all the sessions of the experiment, to mitigate for possible threats arising from the *reliability of treatment implementation*.

Internal Validity: To mitigate the *instrumentation* threat, we conducted a pilot study to verify the design and the instrumentation of our study. The *interactions with selection* threat may affect the internal validity 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 blocking variables related to participants' 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* had significant changes due to the factor **Group**. The effect of this factor is medium with a Cohen *d* value of -0.497 in favor of subjects who play first with the PCG boss and after that with the RCG boss. The subjects in this group (G2, PCG-RCG) demonstrated a greater appreciation for the design of both bosses, both the RCG boss and the PCG boss, than the subjects in the group that carried out the experiment with the other sequence (G1, RCG-PCG). However, both groups value the design of the RCG bosses better than the PCG bosses. The box plots in the bottom center of Fig. 2 illustrate the effects of the factor **Group** and its combination with **Technique** in *Design*. The voluntary nature of participation also poses a *selection* threat, which we mitigated by inviting professional developers and students from a course whose content was in line with the experiment activities to avoid issues with student motivation.

Construct Validity: All of the measurements were affected by *Mono-method bias*. To mitigate this threat we mechanized the measures as much as possible by means of correction templates. The experiment may suffer from the *mono-operation bias* threat since we only compare two representative bosses of each technique.In order to mitigate the *author bias* threat, the tasks were extracted from a commercial video game and the bosses were selected by Kromaia's experts as the most representative of those obtained after the application of the two techniques compared. To weaken the *evaluation apprehension* threat, at the beginning of the experiment, the instructor explained to the participants that the experiment was not a test of their abilities, and that neither participation nor results would affect their grades in the course where the experiment took place.

External Validity: The *interaction of selection and treatment* may pose a threat to our experiment because a different number of participants took part in each alternative of the blocking variables, and players are more represented 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. Other experiments using different games should be performed in the future to further generalise our findings. We have carefully described our methodology and made a replication package publicly available in order to enable other researchers to replicate, reproduce and extend our study.

Table 4: 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
Immersiveness	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

Table 5: Overview of related work. Evaluation: generated content (A), variants of the proposed algorithm (VA), generated content compared to a baseline (C). Measures: Design (De), Difficulty (Diff), Fun (F), Immersiveness (I).

Work	Year	Evaluation	Measures	Hypotheses Formulation	Statistical Analysis	Replication Package	Sample
Cardamone *et al.* [9]	2011	VA	De	✗	✗	✗	5 players
Plans *et al.* [37]	2012	A	F	✗	✓	✗	31 players
Adrian *et al.* [1]	2013	VA	De, Diff, F	✗	✗	✗	22 players
Dahlskog *et al.* [14]	2013	VA	De, Diff, F	✗	✗	✗	24 players
Togelius *et al.* [47]	2013	A	De, Diff, F	✓	✓	✗	147 players
Gravina *et al.* [21]	2015	A	F	✗	✗	✗	35 players
Kaidan *et al.* [25]	2015	VA	De	✗	✗	✗	12 players
Olsted *et al.* [34]	2015	VA	De	✗	✗	✗	13 players
Prasetya *et al.* [40]	2016	C	F	✗	✗	✗	33 players
Ferreira *et al.* [17]	2017	VA	De, Diff, F, I	✗	✓	✗	139 players
Charity *et al.* [10]	2020	A	De, Diff	✗	✗	✗	2 players
Lopez-Rodriguez *et al.* [30]	2020	VA	Diff	✗	✗	✗	30 players
Kraner *et al.* [27]	2021	A	De	✗	✗	✗	5 players
Pereira *et al.* [36]	2021	C	Diff, F	✗	✓	✗	16 players
Brown *et al.* [7]	2022	A	De	✗	✗	✗	35 players
Our work	2024	PCGvs RCG	De, Diff, F, I	✓	✓	✓	32 players + 12 developers

7 RELATED WORK

In this section we describe previous work involving human participants to assess automatically generated video game content, specifically focusing on the empirical elements of their experiments. We refer the reader to previous surveys in the field of automated content generation [23, 49, 59] to learn more about the latest trends and approaches to generate video game content.

Experimentation in Software Engineering is a practice that has been studied for decades [6]. Researchers have adopted established guidelines to be rigorous [57], such as hypotheses formulation, statistical analysis, or including a replication package. However, this has not always been the case for experimentation involving video games engineering.

Video game content generation is a large field [58]. The types of generated content are diverse, such as vegetation [32], sound [37], terrain [19], Non-Playable Characters [54], dungeons [53], puzzles [15], or even the rules of a game [8]. However, it is difficult to find experiments with human participants that compare approaches [3]. Table 5 summarises this work. We observe that previous evaluations involving human participants mainly explore the quality of the content generated by one proposal [7, 47] or different variants of a same proposal [1, 36]. On the other hand, work such as the ones by Pereira *et al.* [35] and by Prasetya *et al.* [40] compared the content generated by their proposal against a baseline (see *Evaluation* in Table 5). Our work is the first that involves human participants to carry out a thorough comparison of two different previously proposed techniques generating content for video games.

In terms of measures, we observe that previous studies have investigated player preferences and perceptions regarding various aspects of video games [43]; this accounts for the use of different measures including design [25, 34], difficulty [30, 35], or fun [37, 40]. Another aspect of video games is the user engagement and immersion, which plays crucial roles in shaping the overall gaming experience [24] (see *Measures* in Table 5). Our work consider all these measures. Previous work have only asked players to evaluate content, i.e., they have not considered the perception of developers (see *Sample* in Table 5). In contrast, we study both the players assessment and the point of view of professional video game developers, and their differences when assessing the quality of the generated content.

Finally, none of the previous works adopt best practices for empirical studies, which are instead widely adopted in general software engineering research. In fact, 73% of the studies have neither hypotheses and validity, statistical analysis, or replication package (see Hypotheses Formulation, Statistical Analysis, and Replication Package columns of Table 5). Our work aims to compare the generated content with empirical rigor. To do so, we adopted the commonly followed guidelines for Software Engineering Research [41].

8 CONCLUSION

Until now, the majority of content generation experiments in game software engineering have failed to conform to best practices for Software Engineering research (*e.g.*, hypothesis and validity, statistical analysis, or replication package). Our research integrates the quality measures embraced by the video game community with the well-established practices of empirical software engineering research. Our results challenge the current dogma by highlighting that content reuse provides advantages towards content generation. Additionally, our findings unlock new possibilities for engaging developers in experimental endeavors. Ultimately, our work can encourage for the empirical game software engineering community to align with the established empirical practices in general software engineering research.

REPLICATION PACKAGE

https://solar.cs.ucl.ac.uk/os/rcgvspcg.html

ACKNOWLEDGMENTS

We would like to express our gratitude to the participants to our experiment, without them this work would not have been possible. The study has obtained ethics approval by Universidad San Jorge (Nº 85/1/23-24) and University College London (UCL-CSREC-207-R). This work has been partially supported by MINECO under the Project VARIATIVA (PID2021-128695OB-100), by the Gobierno de Aragón (Spain) (Research Group T61_23R), by the Excellence Network AI4Software (Red2022-134647-T), by the ERC grant no. 741278 (EPIC) and by the UCL-CS Strategic Research Fund.

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

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

[9] 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.

[10] 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.

[11] 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.

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

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

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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).

[20] 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.

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

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

[23] 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.

[24] 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.

[25] 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.

[26] 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).

[27] 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.

[28] Aristid Lindenmayer. 1968. Mathematical models for cellular interactions in development I. Filaments with one-sided inputs. *Journal of theoretical biology* 18, 3 (1968), 280–299.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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*.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

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

[39] 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.

[40] 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).

[41] Paul Ralph, Nauman bin Ali, Sebastian Baltes, Domenico Bianculli, Jessica Diaz, Yvonne Dittrich, Neil Ernst, Michael Felderer, Robert Feldt, Antonio Filieri, Breno Bernard Nicolau de França, Carlo Alberto Furia, Greg Gay, Nicolas Gold, Daniel Graziotin, Pinjia He, Rashina Hoda, Natalia Juristo, Barbara Kitchenham, Valentina Lenarduzzi, Jorge Martínez, Jorge Melegati, Daniel Mendez, Tim Menzies, Jefferson Molleri, Dietmar Pfahl, Romain Robbes, Daniel Russo, Nyyti Saarimäki, Federica Sarro, Davide Taibi, Janet Siegmund, Diomidis Spinellis, Miroslaw Staron, Klaas Stol, Margaret-Anne Storey, Davide Taibi, Damian Tamburri, Marco Torchiano, Christoph Treude, Burak Turhan, Xiaofeng Wang, and Sira Vegas. 2021. Empirical Standards for Software Engineering Research. arXiv:2010.03525 [cs.SE]

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

[43] 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.

[44] 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

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

[46] Walter F Tichy. 1998. Should computer scientists experiment more? *Computer* 31, 5 (1998), 32–40.
[47] 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.
[48] 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.
[49] 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.
[50] Unity. [n. d.]. Unity. https://unity.com/. Accessed: 01/02/24.
[51] Unreal. [n. d.]. Unreal. https://www.unrealengine.com/. Accessed: 01/02/24.
[52] 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.
[53] 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.
[54] 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.
[55] Steve Watts. 2020. All The Cyberpunk 2077 Delays. https://www.gamespot.com/gallery/all-the-cyberpunk-2077-delays/2900-3618/. Accessed: 01/02/24.
[56] Brady T West, Kathleen B Welch, and Andrzej T Galecki. 2014. *Linear mixed models: a practical guide using statistical software*. Chapman and Hall/CRC.
[57] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. 2012. *Experimentation in software engineering*. Springer Science & Business Media.
[58] Georgios N Yannakakis and Julian Togelius. 2018. *Artificial intelligence and games*. Vol. 2. Springer.
[59] Mar Zamorano, Carlos Cetina, and Federica Sarro. 2023. The Quest for Content: A Survey of Search-Based Procedural Content Generation for Video Games. *arXiv preprint arXiv:2311.04710* (2023).