Explaining the Effect of Procedural Content Generation on Frustration

Dimitrios Christodoulou
Utrecht, The Netherlands
d.christodoulou@students.uu.nl

ABSTRACT

Procedural Content Generation (PCG) in games is a category of algorithms used to generate different kinds of assets, artifacts and behavior while the game is running, either in a loading screen or in realtime. There is some speculation that PCG can lead to unfair game states, which in turn can be a main cause for frustration in players. Frustration is mainly seen as a negative phenomenon, however, on the other end there is evidence that some frustration in games is necessary to ensure that the player does not feel overconfident and also is focused in the game. In this paper, we present a case that investigates how PCG affects frustration and if it leads to overtly negative frustration (and therefore aggression) or helpful and focus-inducing frustration (and therefore a state of flow). An experiment was conducted, and the participants were asked to play some consecutive Pac-Man games. During the experiment, gameplay data, as well as player behavioral data were collected from the players and were analyzed to determine the significance that PCG plays on the apparent frustration of the players. This data was also supported and enhanced by additional information collected by a questionnaire that the users were asked to fill out, which would gauge their feelings on the gameplay experience and whether they felt frustration or if they thought the game was balanced. The results of the experiment showed an increase in frustration when playing games generated through PCG versus normal Pac-Man games. Frustration was not shown to decrease over time, and the main PCG parameters that affected frustration in players were the movement speed of the player controlled character, the inclusion of dead ends in the maze generation process and the duration of the game's powerups.

Author Keywords

PCG; Procedural Content Generation; Games; Frustration in games; Frustration-Aggression hypothesis; Dynamic Difficulty Adjustment; DDA; Player Modelling

INTRODUCTION

It is a fact that games have become a mainstream media outlet and are being adopted by people with all kinds of backgrounds. Each and every one of these people experience games through different lenses and can experience frustration and contentment differently. It is the game developer's job to create a game that is compelling for most, if not all, types and personalities of people. To do that, a balance between frustration and interest needs to be found. In other words, games need to be frustrating enough to be interesting but not so frustrating that players feel overwhelmed and defeated. This is often referred to as the Flow state [12]. In modern games, this need becomes more apparent since many titles use Procedural Content Generation (PCG), a method of dynamically generating in-game content on the fly. With PCG, developers don't need to create complex environments or write entire dialogues, since they can use a PCG algorithm to create content for the game dynamically, while the user is playing or while the game is loading. This method is often used incorrectly in the context of frustration, as the dynamic tuning of the variables can lead to mundane and uninteresting results or, at the other end of the spectrum, results that are so challenging for the player that they become extremely frustrating.

In this paper, a method that generates Pac-Man levels and variable gameplay behavior procedurally is explained and tested. Through the level generation data is gathered to monitor frustration of players due to PCG. To accomplish this, the frustration-aggression theory was used to determine the severity of a response given through frustration by a player. 5 main metrics are used for this purpose:

- **The anticipation of a block by the player.** This is a metric defining if the player is expecting to fail soon.
- **How legitimate the block feels.** Does it feel like the game is doing illegal moves to beat the player?
- **How deliberate the block feels.** For example, does it feel like the game AI makes moves to actively pursue the player?
- **The number of sequential failures to achieve the goal.**
- **The extent of goal completion.** In the Pac-Man game, this metric is defined as the number of pellets (dots scattered in the level) that the player has collected.

Section 2 will cover the above metrics as well as the background of the related areas of research (PCG, the Frustration-Aggression hypothesis, the Flow theory and Dynamic Difficulty Adjustment) and previous related work and the benefits and caveats of the frustration - aggression model and Flow theory. Section 3 will describe the hypotheses of the paper, the methods used to produce this paper's results, the produced game and the experiment procedure, as well as the data processing necessary to confirm the proposed hypotheses. Sections 4 and 5 will present and discuss the obtained results and describe future work that can be done to further investigate and improve upon the results of this paper.

BACKGROUND AND RELATED WORK

Procedural Content Generation

Shaker, Togelius and Nelson [17] define PCG in games as the automatic or computer-assisted generation of content such as levels, landscapes, items, rules and quests. PCG has many benefits, such as randomness for unpredictable gameplay, smaller file sizes and larger amounts of content without taking up large amounts of developer time.

There is already plenty of research and commercial work on PCG. It is not only used in games, but in most creative areas, as it has, for example, been used to generate music or paintings. It is also used in multiple areas of games and game development and this paper focuses on this aspect of PCG. PCG in games (also frequently called PCG-G) is used to generate level layouts, world terrain, weather conditions[3], Non-Player Characters, their traits and behavior as well as dialogue, quest and storyline scripting [11] and, in some cases, gameplay mechanics generation.

In the paper "Procedural Content Generation for Games: A Survey" [8], the authors create a complete taxonomy of game elements that can be procedurally generated. These elements can range from small game bits, like sound effects, to individual game systems, like road networks, and finally to game design elements, such as an overarching procedurally generated world design. The authors also show numerous examples of commercial games that implement PCG in the above contexts.

Lopes and van der Linden [21] discuss in-depth the different ways that PCG is used in dungeon generation and their pros and cons. In another paper by Pedersen, Togelius and Yannakakis [14], gameplay data is divided into two categories, controllable features and gameplay features. The first category describes the parameters used to generate game levels, while the second characterizes player behavior. PCG overall helps with controlling and modifying controllable gameplay features, while player modelling is the area best suited to utilize and exploit player behavior.

In an entirely different approach, Stein et al. [19] measure the mood and excitement of players via electroencephalography signals. By quantifying this excitement and putting it into formulas the authors define a PCG system that generates behavior and new levels based entirely off of biological signals.

Frustration in games

At the same time, there is research that tries to connect which game elements produce feelings of contentment or frustration in people when playing games [22]. A lot of formulas have been brought forth to explain this connection, with the most prevalent ones being the Frustration - Aggression hypothesis first proposed by Dollard et. al. [6] and then reformulated by Berkowitz [2] and another one being the Flow theory by Csikszentmihalyi.[5] [12].

Frustration - Aggression Hypothesis

Dollard [6] proposes that frustration is "an interference with the occurence of an instigated goal-response at its proper time in the behavior sequence". What this means is that frustration arises as a block when headed towards a specific goal. According to Berkowitz, frustration can give rise to aggressive inclinations because they are aversive [1]. When combined with different variables, this frustration can lead to aggression. These variables range from environmental aggressive cues, to previous experience with frustration and aggression, to personality and temperament, social and cultural background and the availability of alternative responses to stimuli, such as excitement. The last one is the most important when contextualizing the frustration-aggression hypothesis in the context of games, as games often provide alternative emotional responses to mitigate the effect of frustration on players.

Berkowitz then moves on to explain how frustration happens. Frustration is apparent only when a person has been blocked from reaching a particular goal, and that goal has the potential to provide meaningful rewards to the person. When the anticipation of the potential reward is stronger, the feeling of frustration is also stronger and the more likely it is to translate to aggression. Also, when the blocking of the goal is limited, so too is the feeling of frustration limited. Berkowitz then adds to this theory a few more factors that can affect the strength of the illicited response. The first factor is how legitimate the block is, with illegitimate blocks causing much more frustration than otherwise. The second factor is how deliberate the block feels to the player. In this case blocks that are seen as accidental cause less frustration than deliberate-looking ones. Furthermore, when a person expects to fail and be blocked there is a significant mediation of the frustration feeling, while at the other end, unexpected failures correspond to much greater feelings of frustration. Finally, quickly repeated failures have a much greater impact on frustration according to Berkowitz, when compared to failures that are significantly apart time-wise.

Flow Theory

The Flow theory, first suggested by Csikszentmihalyi[4] supports that there exists a mental state of optimal experience, a state characterized by focus and intense attention and involvement in one's activity. While in this state, people experience an increase in their perceived skill and chances of success in their activity [12]. This state is called flow because the person's actions in this state flow from one action to the next seamlessly and unconsciously. While in Flow, the person loses their sense of self and their attention lies solely in the performed activity.

Specifically in games, players experience flow when their skills and the game's difficulty increase at an equal and similar fashion. When looking at this theory from a game design perspective, any deviation in the game's difficulty can result in the loss of flow by the player. This is much more prevalent when PCG is used to tune the behavior of the game, because the inclusion of randomness can lead to difficulty extremes and can therefore break the experience of flow.

Dynamic Difficulty Adjustment

Dynamic Difficulty Adjustment (DDA) is a mixture between AI, Game design, PCG and the field of psychology in games. The main point of DDA is to adjust the difficulty of a game dynamically so that it corresponds, on average, to the skill level

of the player. Many attempts have been made to categorize DDA methods to different taxonomies based on game styles [7] [16] or on used algorithmic approaches [13].

Most methods use AI to modify the behavior of the game agents and make them increasingly more difficult, like the implementation of a DDA method in MOBA games from Silva, Silva and Chaimowicz [18].

Other methods explore the DDA space by interacting and modifying the game environment. In a paper by Robin Hunicke [9], the author explores a method called the "Hamlet system", which tracks player behavior and estimates the player's probability of death by the usage of the items in their inventory (the gameplay mechanic where all the items of the player are stored). It then increases or decreases the amount and quality of items found based on the difficulty the player is facing. All this is done to keep the player inside a "comfort zone", where the probability of death is neither high nor low.

Yet other approaches use a more formulaic method and apply DDA by tweaking variables to achieve easier or more difficult game levels, as seen in the paper by Rhio Sutoyo et al. [20]. In the above paper, the authors performed DDA on a tower defense game by having a difficulty modifier, which affected all gameplay variables on the level, such as enemy health and speed.

Finally, there are methods that alter a game's difficulty by procedurally generating new and different level layouts. In a paper by Jennings-Teats et al. [10] the authors introduce *Polymorph*, an algorithm that generates 2D platformer levels, by incorporating a statistical model that tracks the overall difficulty of the so far generated level. By combining the above with a metric gauging the skill of a player the algorithm ranks the level segments that can be generated by difficulty and picks the one most appropriate.

All of the above approaches use PCG in some capacity; whether to randomize AI-driven behavior, or environmental settings, or to pseudo-randomize modifiers used in formulas or finally to generate new level layouts altogether. It is therefore clear that PCG and DDA are connected and exploring new methods in one of these fields can help with progressing in the other.

Contributions of this paper

This paper contributes to the study of the role that PCG plays on player behavior by measuring frustration. Player behavior was modelled inside a Pac-Man game by gathering different metadata about the player, some of which are used to calculate frustration. By including PCG and monitoring this data, we measure the impact that PCG has on frustration and boredom and pave the way for future research to expand or go more in-depth on this or similar metadata.

By administering the experiment online in a semi-controlled environment, we exclude any physical parameters from the data collection due to the difficulty of obtaining accurate data conforming to the real-world scenarios that the participants will perform the experiment in.

METHODOLOGY

Overview

The presented paper aims to improve our understanding of how randomization affects frustration (and its counterpart, boredom) in games. In this section, a high-level overview of steps, describing how we plan to analyze this is presented and explained. **Step 1:** online participants will take part in an experiment that will help us gather data on PCG and frustration metrics. Each participant will play four Pac-Man games, while data is collected in a database at set intervals of five seconds. **Step 2:** this data will be pre-processed to enable further statistical analysis. In this step it is important to remove irrelevant data (such as players who have not played the experiment in its entirety), so as not to skew the data. **Step 3:** From the final dataset we will try to analyze which variables play a greater role in frustration than others, by performing a significance analysis on the variables, through appropriate statistical methods (ANOVA and a Generalized Linear Model - GLM).

In the presented experiment randomly selected participants are called to play four Pac-Man levels, one without randomization and PCG and three with randomized gameplay behavior and level layouts generated with PCG. The main goal of the experiment is to describe how and at what degree PCG affects frustration in players.

Before the game implementation is discussed in depth, there needs to be a clear idea about what the research questions we hope to answer are. In total, four research questions are presented which this study will try to answer.

Key Research Questions

- **Research question 1**: Is there a difference in frustration between levels generated through PCG and normal game levels?
- **Research question 2**: How much does randomization and the unfamiliarity with the level layout affect a player's frustration?
- **Research question 3**: Will players become more accustomed to PCG toward the end of the gameplay session?
- **Research question 4**: Which behavioral or layout variables, when created procedurally, affect the frustration emotion the most?

The Game

The Game Implementation

To test the above questions, a custom version of the Pac-Man game was created, with logic to generate custom levels procedurally. Pac-Man is a game where the player controls the eponymous character through a maze, with the goal of collecting all of the pellets in the level to win. Wandering around the maze are also 4 ghosts which seek Pac-Man. If any ghost touches Pac-Man, then the game is over. The player loses a life and has to replay the level, while keeping the collected pellets. The player has 3 lives in total at every playthrough. This means that every death does not reset the game entirely. Also

scattered across the maze are power pellets. These one-use pickable items frighten the ghosts when picked up and allow Pac-Man to eat them and help the player score more points. This specific game was chosen since most people are familiar with the game or at the very least have heard of it. It is also a game which requires low effort to grasp and obtain a minimal level of proficiency with, which will provide the players with a much better understanding of when they feel frustration. The foundation for the game was built by previous research [22] and most, if not all, of the implementation specifics were obtained by the Pac-Man dossier [15], an extensive explanation and background research on Pac-Man's implementation details.

This Pac-Man implementation was a custom-built version of Pac-Man developed by the researcher in Unity 2021.3 specifically for this research. The Pac-Man build was, overall, identical to the traditional Pac-Man game, with minor differences.

Before modifying game elements procedurally, we need to alleviate some of the extremes that can be prevalent in procedurally generated levels. First, it is necessary that when ghosts are frightened and eaten by Pac-Man they do not move back to the original box, since when combining this with the altered level layout this can lead to extreme behaviors and it also introduces inconsistent frightened durations for ghosts. The latter happens because, when eaten, ghosts' frightened timers are reset. To alleviate the above, the mechanic which makes ghosts go back to their initial position after they are eaten by Pac-Man, while he is under the effect of a power pellet, to reform was removed. It was replaced with a mechanic where the ghosts would reverse their direction and move towards the other side of the map and away from Pac-Man. This ensures a consistent time where ghosts are frightened. At the same time, picking up a second, third or fourth power pellet, while the duration of the previous one is not expired will refresh the duration to the duration of the newly picked up one.

To check the impact of PCG in player frustration as well as answer the main presented research questions, the paper presents a PCG engine that randomizes specific game elements. First, the main points of the core gameplay loop that could cause frustration when generated in a PCG context were identified. Essentially, these are game elements that, when modified procedurally, cause the greatest changes in gameplay. These elements were the ones modified, and are highlighted below, in order of importance on gameplay.

Implemented PCG Elements

1. **The connecting side-corridors**. These allow Pac-Man to travel from one side of the screen to the other and are shown as side corridors that extend outwards away from the main level. Initially, their location is static and always at the middle-left and middle-right side of the level.

   By modifying their locations, we introduce unpredictability and longer travel times between teleportations. So, their location was randomized on the entirety of the left and right walls, following a rule stating that the entire corridor should be placed without cutting any of the corner topology of the level.

2. **The location of power pellets.** The locations of these four pellets were randomized and possible locations spanned every position where Pac-Man can travel to. This was done to create different travel times, leading to frustration or boredom.

3. **Pac-Man's starting location.** A location is selected at random from all available free spaces in the maze and Pac-Man spawns there. This enhances game repeatability by increasing unpredictability and introducing longer initial travel times.

4. **The Ghosts' speed.** In the randomized version of the game, ghosts can move at a range of -50% to +50% of their unmodified speed value (the value present at the base game without any PCG.)

   By modifying their speed, we again introduce unpredictability, which can lead to increased frustration when the ghosts move faster or increased boredom when they move slower.

5. **Pacman's speed.** For similar reasoning as with the ghosts, Pac-Man's speed is modified randomly in a range from -50% to +50% of its original unmodified speed.

6. **Each ghost's appearance time.** Every ghost comes out of the spawning box after a specific interval (8 seconds in the non-PCG version of the game). The implementation randomizes this interval in the range between 6-10 seconds to affect the level of frustration caused by many (or all) ghosts spawning at frequent intervals (see figure 1 below)

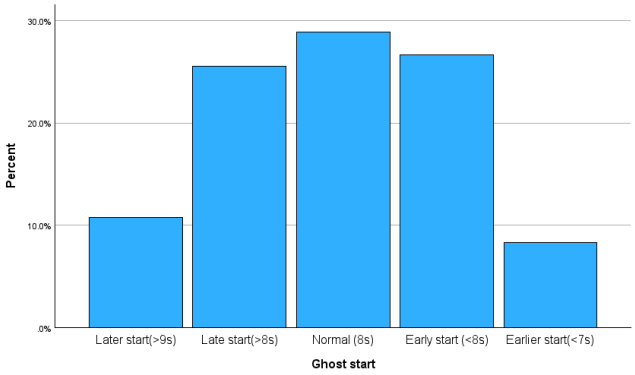
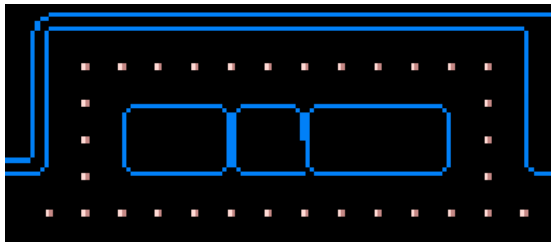


Figure 1: *The distribution of the randomization of the ghost appearance timer.*

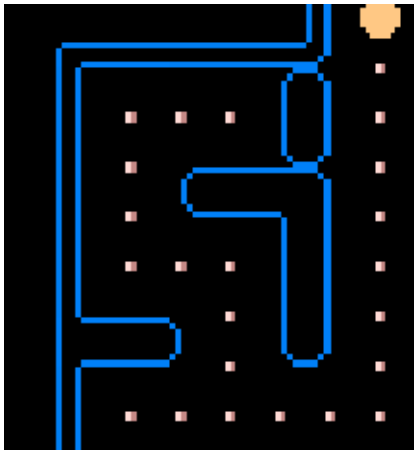
7. **Scatter and chase durations.** In the original Pac-Man game, the ghosts have 2 potential AI states; a scatter mode and a chase mode. In the scatter mode, each ghost moves in a corner of the maze reserved for it at the beginning of the game. This allows Pac-Man to move through the maze more easily for some time. In the chase mode, the ghosts actively follow Pac-Man by reading its location. The ghosts in the original game begin their AI with a 7-second scatter mode followed by a 20 second chase mode. This pattern is repeated 2 times, followed by a 5 second scatter and 20 second chase pattern and then finally the ghosts enter a 5 second scatter and then enter chase mode for the

remainder of the game. We will refer to the 7 second scatter as the 1st-part scatter and the 5 second scatter as the 2nd-part scatter. This randomized version of the game includes random durations for the scatter (both 1st and 2nd part) and chase state per ghost.

8. **Level layout**. To also test the effect of applying PCG in a level generation context, the testbed game includes changing the layout of the level. This is done by walling off certain areas of the maze at the level generation stage. There are two versions included in the built game. The first is a version that walls off an entrance in an area with 3 or 4 entrances. This allows Pac-Man to traverse every area of the maze without backtracking. The second version introduces dead ends to the maze, where the players will need to backtrack to return to the active game area. The reason these two methods are not combined is that by including dead ends there is the possibility that a ghost will follow Pac-Man inside a dead end and therefore the player will have no way to escape. We felt that it is important to keep this option separate for easier handling of cases where we would like to disable such a randomization. However, to better test how significant this mechanic is on frustration, we decided to keep dead ends in the experiment.



(a) *A walled-off area that's not blocking Pac-Man's movement.*

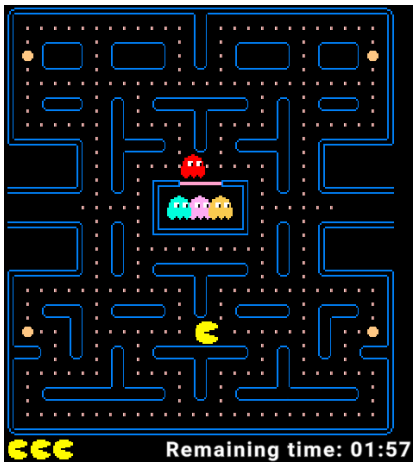


(b) *A walled-off area that creates a dead end.*

Figure 2: *Level layout procedural generation.*

In addition to the above, a timer was introduced that, when expired, will cause Pacman to die and the player to lose one of their lives. This was added to produce additional cause for frustration in order to measure the effect of time-based constraints on frustration. This has the added benefit that the participants have an upper bound on how much time the experiment will last. The timer is set to 120 seconds per level, leaving ample time for players to collect all pellets and clear the level.

The game that was produced as part of the research experiment randomizes the above elements, by using noise as a random generation engine. A seed is generated at the start of a level and is then used to generate the entire level layout as well as randomize the above behavioral values for Pacman and the ghosts. This means that by entering the same seed as input, the resulting level will be always the same. This is done to ensure that the results are consistent between players and reproducible. Creating a persistent level is also a requirement for a good PCG engine.



(a) *A level generated without any randomization. The location of the power pellets and side corridors is static and symmetrical. Pac-Man's starting location is in the middle of the level.*

(b) *A level generated with randomization. The location of the power pellets and side corridors is asymmetrical. Pac-Man's starting location is in a random spot in the level.*

Figure 3: *Level layout procedural generation.*

Game Element	Type of Value	Possible Values
Connecting side-corridors	Starting coordinates of corridor	∼56 different corridor layouts (∼28 per corridor)
Location of power pellets	Coordinates	∼288 different locations (number of empty tiles in the maze)
Pac-Man's starting location	Coordinates	∼288 different locations
Ghosts' speed	Numeral	50-150% of original base value
Pacman's speed	Numeral	50-150% of original base value
Ghost appearance time	Numeral	6-10 seconds
Scatter durations	Numeral	5-9 seconds in part 1, 3-7 seconds in part 2
Chase Durations	Numeral	18-22 seconds
Level Layout	Set of coordinates	∼848 different layout combinations

Table 1: *A list of randomized game elements, their value types and their possible values*

EXPERIMENT

To ensure that a wide audience of different age groups, backgrounds and levels of gaming experience was reached, the experiment was administered online. The game was built as a WebGL Unity game and published on Unity's official Play platform, where creators can freely publish WebGL builds. To store the player responses and the values of the PCG algorithm, a database was created.

This database is a MySQL database hosted in a cloud environment through Amazon RDS (Relational Database Service). To ensure a secure database connection a server side PHP script was created to act as a proxy between the game and the database. The script acts as an API and exposes an endpoint that inserts the data to the database.

Figure 4: *The main screen that the player sees when starting the experiment. Games 2,3 and 4 are greyed out and are "unlocked" sequentially as the player progresses. This was done to ensure a smooth flow of the experiment.*

Experiment Setup

A participant will need to follow a few steps in order to complete the experiment. **Step 1:** First the participant is presented with an overview screen mentioning the basics of the experiment and then with a consent form that informs them that they can exit the experiment at any time. **Step 2:** The subject then views instructions on how to play the game and what flow to follow. **Step 3:** They are then asked to play the Pac-Man game one time to introduce the game to subjects who haven't played it and help others to re-familiarize themselves with the movement and layout of the game. It will also help create a benchmark upon which frustration components due to PCG will become more apparent. **Step 4:** After this initial playthrough, the candidates will need to complete a questionnaire with some demographic and gameplay questions. **Step 5:** After this, participants will play the game again 3 times but with randomization and PCG enabled. After each playthrough, the candidates will need to fill out the questionnaire again (without the extra demographic questions, which are necessary only once).

Collection of Participants

Participant selection was performed randomly from various sources in order to ensure a diverse sample regarding at least gender, age group and familiarity with gaming. Such sources include the Utrecht University forums and groups and, since subjects there were expected to be at the age group of 18-28, the experiment was posted in other groups with a higher expected mean age. As a further contingency, for the case of low participation rate, additional participants were selected among friend and relative groups of the researcher.

It was expected that around 5-10% of the total participants would not complete the experiment fully, due to time limitations (even though the experiment takes at most 15 minutes), technical issues or by closing the browser window before the experiment is completed. In these cases, results from those participants were filtered out.

Data Collection

To ensure that there is sufficient amount of data to reach an adequate conclusion and answer the above hypotheses, a data collection mechanism stores gameplay data at set time intervals, as well as at the time of Pac-Man's death. The interval between data storage is set to 5 seconds.

The data that is stored is a mixture of gameplay, layout and behavioral variables. Some of the measured frustration variables

were calculated according to Berkowitz's [1] definitions on the effect of blocks on frustration. The overall data gathered is the following:

- **The seed of the level** - This is the initial seed that is used to generate the level layout. It is an indicator of an individual game level, since it is unchanging throughout a level. It is also used to view the level layout and what level was generated from the PCG engine.
- **Level type** - This variable is used to discriminate whether the level is a normal one or a PCG generated one. The value is set to either "PCG" or "Normal" depending on the type of level the player is playing.
- **Data Timestamp** - The time that the data was inserted to the database. We use this to verify continuity of gameplay and also potentially use it for time series analysis.
- **Ghost Appearance Rate** - In a normal Pac-Man game, ghosts appear at a rate of 8 seconds per ghost. However, in PCG games ghosts have a wider appearance rate. Therefore, apart from the value of 8 seconds, every other value indicates the presence of randomization in the level generation.
- **Scatter Duration** - This indicates how much time will the ghosts spend on their scatter behavior. This variable is used in calculating how much time ghosts spend chasing or moving away from Pac-Man and is subsequently used to calculate *block deliberateness* and *block anticipation*.
- **Power Pellet Duration** - How much time do power pellets affect the player for. This is set to a value of 8 seconds in the normal game, but in PCG games it is randomized in a range from 6 seconds to 10 seconds.
- **Left Teleport Location** - The starting coordinates of the left teleport side-corridor.
- **Right Teleport Location** - The starting coordinates of the right teleport side-corridor. All gathered coordinates have no statistical importance, but would potentially be used in a frequency calculation to determine if a particular location was significantly more frustrating than others.
- **Dead Ends** - Whether dead ends were included in the level generation process. This is an important factor because it enables associating the existence of dead ends with frustration increase or decrease.
- **Pac-Man position** - The starting coordinates of Pac-Man. It was proposed that during a PCG game, if Pac-Man would start close to the center of the level, frustration would increase and vice-versa, so this variable was included to determine if this is the case or not.
- **Power Pellet Locations** - The starting coordinates of all the power pellets.
- **Time played** - How much time measured in seconds, did the player take to finish the level (either by winning or losing). This variable was also used to help with determining continuity in the level (i.e. that the player did not wait for the level to complete without doing anything).
- **Remaining Lives** - How many lives (3, 2 or 1) does the player still have at the time the data is inserted in the database. This, together with the collected pellets and power pellets (below), are important factors to the frustration calculations, since they determine how close the player is to completing or failing the level.
- **Collected Pellets**
- **Collected Power Pellets**
- **Ghost Speed** - Both the speed of ghosts and Pac-Man's speed (below) play an important role in the balance of the level and need to be collected in order to verify their significance on total frustration.
- **Pacman Speed**
- **Total time spent in scatter mode** - As mentioned already, this variable in used to calculate *block deliberateness* and *block anticipation*.
- **Extent of goal completion** - This measures how close the player is to completing the level. It is essentially a sum of the collected pellets, power pellets, passed time and time spent remaining still (due to being stuck on a wall).

  It is calculated through the formula

  $$EGC = \frac{CP}{240} + \frac{CPP}{4} - \frac{TA}{240} - \frac{TSM}{100},$$

  where CP is the amount of collected pellets, CPP the amount of collected power pellets, TA the time alive and TSM Pac-Man's time spent moving.
- **Number of sequential failures** - This is a metric that shows the effect that multiple quick failures have on frustration. It is calculated through the formula

  $$NSF = -\frac{ND}{3} - timeBeforeDeath,$$

  where ND is the number of deaths.
- **Block anticipation** - Measures how legitimate the block of reaching the goal state feels to the player. The formula used to describe this is

  $$BA = -AGD - NGD - AGS,$$

  where BA is the block anticipation, AGD is the average ghost distance from Pac-Man, NGD is the nearest ghost distance and AGS is the average ghost speed.
- **Block deliberation** - Measures how deliberate the block of reaching the goal state feels to the player. The formula for calculating the metric is

  $$BD = \frac{(TSM+TSC)}{TST},$$

  where BD the block deliberation, TSM the total time spent moving toward Pac-Man (while not chasing), TSC the time spent specifically in the chase mode and TST the total scatter time (time spent in scatter mode).

- **Total Frustration** - The dependent variable of the experiment. It is measured as

$$TF = EGC + NSF + BA.$$

Demographics and Player Perceptions

In addition, due to the necessity to include players' personal opinions on the procedurally generated levels, an in-game questionnaire was created. The questionnaire would pop up after a player would win or lose a level. The first time the questionnaire pops up it also asks some demographic questions to more accurately categorize the effect of PCG on age groups and game familiarity. Specifically, during the first game (the one identical to the original Pac-Man game), participants were asked:

- Minor demographic data (gender, age range and familiarity with games)
- How they felt about the overall challenge of the level.
- Whether they found the controls frustrating.

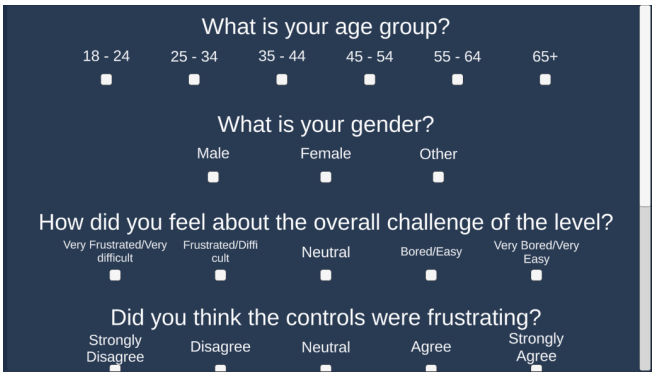


Figure 5: *Part of the questionnaire that the player sees when finishing the first level.*

In the second, third and fourth games (the ones that were randomly generated through PCG), participants additionally were asked what they thought about the generated PCG layout. All answers to the above questions were in the form of Likert scale, ranging from 1 (Strongly Disagree) to 5 (Strongly Agree). The questions were formed this way to measure a response in conjunction with both the frustration-aggression hypothesis and the flow theory. Answers corresponding to strong feelings of frustration are categorized as closer to aggression, while answers corresponding to medium or mild frustration are categorized as closer to a state of flow and support the hypothesis of positive and focus-inducing frustration.

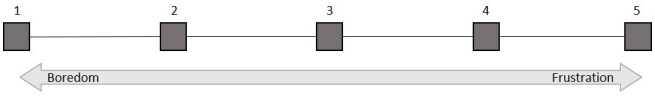


Figure 6: *Answers closer to 1 in the likert scale questions are tied more closely to boredom, whereas answers closer to 4 are tied to the state of flow. Answers of 5 in the scale are tied to feelings of aggression.*

Data Processing

As mentioned, each participant had to play 4 games, one identical to the original Pac-Man and 3 modified through PCG. Each game recorded snapshots of gameplay roughly every 5 seconds. This resulted in a large dataset which needed to be filtered before performing a statistical evaluation model. Initially, some outliers were removed, as corrupt or missing data. Then, all variables containing data specific to each ghost were combined to represent mean values for all 4 ghosts (effectively representing the ghosts' behaviour as a group entity). Afterwards, all confounding variables were removed (ones that determine another variable) and replaced with the variable they determine.

Once data from the experiment was gathered and preprocessed, it was fed into a Generalized Linear Model (GLM). This model will determine the significance of the different randomized elements selected through a backwards selection process (a repeated process where the least significant elements are removed until all the remaining variables are significant enough to be kept in the model).

RESULTS

Exploratory Analysis - Experiment Demographics

In general, a total of 66 people participated in the experiment generating approximately 3,100 "raw" gameplay data. Out of these participants, 4 were removed because of incomplete game data. Of the remaining participants, 68.3% were **male**, 28.5% **female**, and 3.2% stated their gender as **other**.

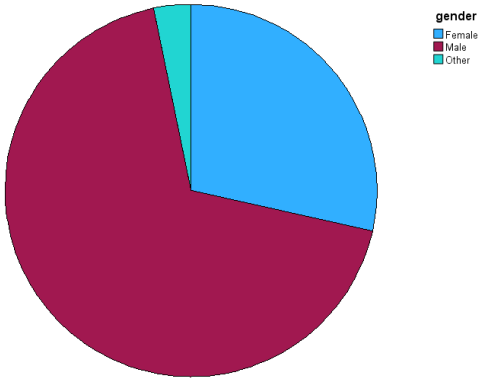


Figure 7: *Gender Distribution*

People who participated in the experiment ranged from different age groups, with 20.5% between 18-24, 50.6% between 25-34, 16.1% between 35-44, 4.8% between 45-54 and 6.4% between 55-64. One participant was over 65.

Finally, a large percentage of people were already at least partially familiar with games and played regularly during the week, as is now becoming the norm. 13.4% stated that they never play games, while 28% state that they play at most 2 hours per week. From that point, 29% play between 2 and 8 hours every week, 21.5% play between 8 and 20 hours every week and 8.1% stated that they play at least 20 hours per week.

The above distributions provide a diverse set of participants from different age groups and with different gameplay behaviours.

Exploratory Analysis - Gameplay perceptions

Participants were asked to fill in a questionnaire between each game session documenting their experience with that particular level layout.

Regarding the controls of the game, 60.7% of the participants thought that the controls were frustrating, 17.7% thought they weren't frustrating and 21.5% were neutral. On the subject of perceived level difficulty, results tend to very slightly favor the "easy" side of the spectrum. 29.6% found the gameplay boring and easy, while 24.2% found the game to be frustrating and difficult. 46.2% were neutral and had no opinion on the gameplay difficulty. These perceived results can also be attributed to the participants strong background and familiarity with games in general.

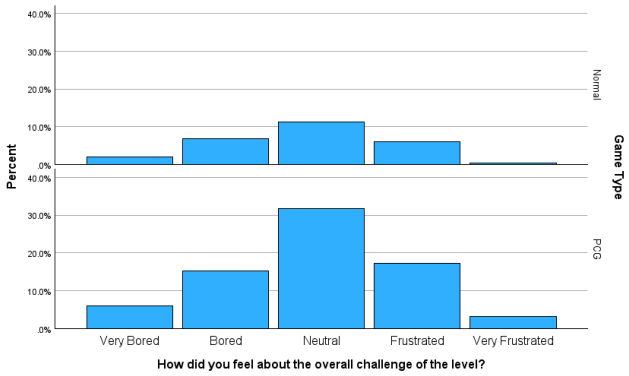


Figure 8: *Answer distribution about perceived difficulty.*

Finally, PCG and the overall interference on the level generation parameters and the gameplay balance variables tend to go towards the more frustrating side, but with the majority of answers remaining neutral at 72.1%. A 12.4% of answers stated that the players were bored or felt that the PCG changes were uninteresting and a 15.5% felt the opposite, meaning that the gameplay was more difficult and/or frustrating.

Statistical Analysis and Experiment Results

A two-step analysis was performed to check the results of the experiment. First, a one-way ANOVA was performed between the two samples (levels identical to the original Pac-Man and levels modified through PCG), with the dependent variable being in this case the total frustration of the player as explained in the Experiment section.

In the test, a clear difference between normal and PCG mean levels can be seen, with normal levels reaching a mean of 21.1 and PCG levels increasing the total frustration to 22.27, a difference highly significant at level >99%. This confirms the **first research question** posed above, meaning that static game levels are less frustrating than randomized ones.

Descriptives

	N	Mean	Std. Dev.	Std. Error	95% CI for Mean		Min	Max	Between Comp. Var
					Low	High			
Normal	795	21.07	6.54	0.23	20.62	21.53	5.24	45.66	
PCG	1962	22.30	7.81	0.18	21.95	22.65	3.41	47.09	
Total	2757	21.95	7.49	0.14	21.67	22.23	3.41	47.09	
Fixed Effects			7.47	0.14	21.67	22.23			
Random Eff.				0.66	13.58	30.32			0.70

Table 2: *PCG and Normal level descriptives by ANOVA*

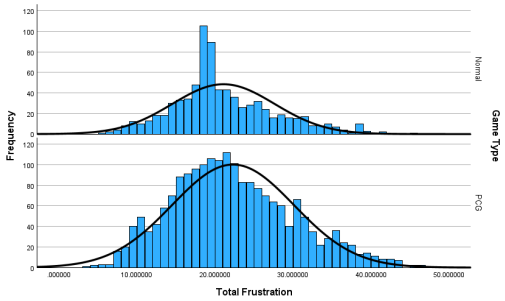


Figure 9: *Frequency plot of frustration between Normal and PCG games. PCG games have a higher mean total frustration.*

ANOVA

		Sum of Squares	df	Mean Square	F	Sig.
Between Groups	(Combined)	850.02	1	850.02	15.25	0.000
	Linear Term	850.02	1	850.02	15.25	0.000
Within Groups		153602.28	2755	55.75		
Total		154452.3	2756			

Table 3: *ANOVA results between PCG and normal levels*

Furthermore (not shown in the ANOVA table), an accompanying analysis of homogeneity of variances between normal and PCG groups indicated significant differences in variances, with the PCG group exhibiting higher variability. This is also visible in figure 9.

After the significance in difference between the 2 groups of data was established, the next step was to understand which variables contributed the most to frustration. For this goal, a Generalized Linear Model (GLM) was implemented. Initially a full model was built, a model with all explanatory variables, demographics and perceptions. Afterwards and iteratively, not significant variables were discarded through the use of *backwards selection strategy* with a threshold of 95% significance (F-value). For reasons of simplicity, no interaction effects were included in the model, only main effects. A total of 11 GLM models were sequentially implemented in order to discard non-significant variables with the remaining ones being those that contribute the most to the player frustration component. During the backwards selection strategy, the variables that were discarded from the model and their significance (F-value in parentheses) were, in the order of removal:

block deliberation (0.914), gender (0.903), perceived control (0.69), game duration (0.696), perceived intensity of the PCG generation (0.697), collected power pellets (0.450), ghost start time (0.462), ghost speed (0.124), frequency of gaming hours per week (0.068) and age (0.065). The two latter variables are borderline significant (significant at 90% level).

The variables that remained in the final model for frustration are shown in table 4 below.

	Likelihood Ratio	df	Sig.
(Intercept)	5064.931	1	0.000
Power pellet duration	9.427	2	0.009
Dead ends	8.427	1	0.004
Remaining Lives	7.860	2	0.020
Perceived Difficulty	7.601	2	0.022
Collected Pellets	61.002	1	0.000
PCM speed	12.512	1	0.000

Table 4: *Final table of variables and their significance on the experiment after all GLM iterations.*

The final model, seen in table 5, (the one with no possible removal of explanatory variable) includes both variables that explain the effect of PCG (group 1) and variables that influence frustration and whose effect must be removed (group 2).

The most significant contributors to frustration in Group 1 in the experiment were Pac-Man's speed (significance below 0.001), the existence of dead ends in the generated level layout (0.004) and the duration of the power pellets (0.009).

Important variables in Group 2 include the player's remaining lives (0.02), the collected pellets (significance below 0.001) and the player's perceived difficulty of the game level (0.022), which is the answer the player gives to the question regarding how challenging they felt the level was.

Important note: For the interpretation attempted below, please note that frustration values (initially negative) were inverted to positive, therefore smaller values signify higher frustration.

The model shows that each increase of Pac-Man speed by 0.1, results in a decrease of frustration by 2.25. Similarly, each 100 collected pellets increase frustration by 1.4, apparently due to the feeling of reaching the goal.

Regarding nominal variables, dead-ends increase frustration by almost 0.9, while longer or shorter power pellet duration reduces frustration. Remaining lives have an even more prominent effect on frustration, with each decrease of lives from 3 to 2 and 2 to 1 resulting in higher frustration (again possibly due to the feeling of reaching the goal).

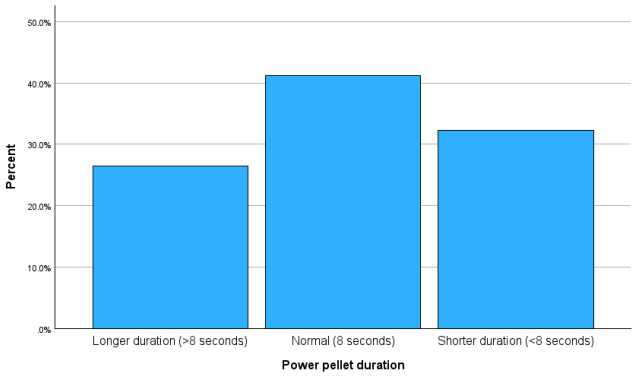


Figure 10: *The distribution of the randomization of the duration power pellets.*

DISCUSSION

Can PCG modify frustration in games?

The initial goal of this experiment and research was to find out how PCG affects frustration, namely which elements can be tuned to modify the severity of frustration responses by the player. According to the experiment results, PCG seems to affect frustration and excitement, even when some parameters are adjusted to assist the player. The significance of inclusion of dead ends in the final model, as well as the duration of power pellets seems to suggest a direct correlation between frustration and the elements that directly and immediately influence the player's freedom in the game. This can also be further proved by the non-significance of variables that do not seem to directly affect the player, such as the speed of the ghosts or the location of the teleportation corridors.

Through the **fourth research question** posed at the start of this paper we try to understand which PCG and behavioral variables affect the frustration component the most. From the GLM model, it is apparent that there are 3 variables that show the effect of PCG in frustration. *Pac-Man's speed* aids the player and leads to a more relaxed game, while the inclusion of *dead ends*, as mentioned, severely limit the ability of the player to navigate the maze and therefore increase the frustration component. On the other hand, the *duration of the power pellets* is more important as the player familiarizes themselves with the game and gets a feeling of how long the "frightened" effect lasts.

Is frustration modified by PCG negative or positive?

As mentioned earlier, frustration can be conceptually positive when it is focus-inducing, as is suggested by the Flow Theory [4], or on the other hand it can be overtly negative, leading to aggression and the feeling that the player is deliberately being targeted. In this case, the results from the experiment show that the induced frustration from PCG tends to be more focus-inducing than aggression-inducing. This can be verified by the ANOVA means difference. It is clear that the mean frustration during normal games was around 21.1, while in PCG modified games the mean was fluctuating around 22.27. The two values are close to each other and the fact that no extremes show

Parameter Estimates

Parameter	B	Std. Error	95% Wald Confidence Interval		Hypothesis Test		
			Lower	Upper	Wald χ^2	df	Sig.
(Intercept)	20.151	1.6113	16.993	23.309	156.399	1	0.000
[Power pellet duration=1.00]	-0.675	0.3829	-1.426	0.075	3.110	1	0.078
[Power pellet duration=2.00]	-1.055	0.3443	-1.730	-0.381	9.398	1	0.002
[Power pellet duration=3.00]	0[a]						
[Dead ends=0]	0.895	0.3080	0.291	1.498	8.439	1	0.004
[Dead ends=1]	0[a]						
[Remaining Lives=1]	-1.157	0.4602	-2.059	-0.255	6.319	1	0.012
[Remaining Lives=2]	-0.774	0.3602	-1.480	-0.068	4.620	1	0.032
[Remaining Lives=3]	0[a]						
[Perceived Difficulty=1.00]	-0.430	0.3921	-1.199	0.338	1.204	1	0.272
[Perceived Difficulty=2.00]	-0.932	0.3468	-1.611	-0.252	7.214	1	0.007
[Perceived Difficulty=3.00]	0[a]						
Collected Pellets	-0.014	0.0018	-0.018	-0.011	61.682	1	0.000
PCM speed	22.545	6.3664	10.067	35.023	12.541	1	0.000
(Scale)	52.689[b]	1.4191	49.980	55.545			

Model:
Frustration = (Intercept), PP dur., Dead ends, Rem. Lives, Perc. Diff., Collected Pellets, PCM speed
a. Set to zero because this parameter is redundant.
b. Maximum likelihood estimate.

Table 5: *Final GLM parameter estimates*

in the PCG frustration component proves that the component is not so high so as to reach an extreme level and therefore lead to aggressive responses. This is also apparent from the perceived difficulty of the PCG generated levels. Players usually felt neutral or a bit more frustrated, but not strongly frustrated (only 1.1% felt that the game was very frustrating), meaning that no aggression component was apparent during most experiment runs.

The above facts answer the **second research question** initially proposed, meaning that randomization of the game level and of the various behavioral components do not tend to affect frustration to an extreme.

Frustration component analysis over time

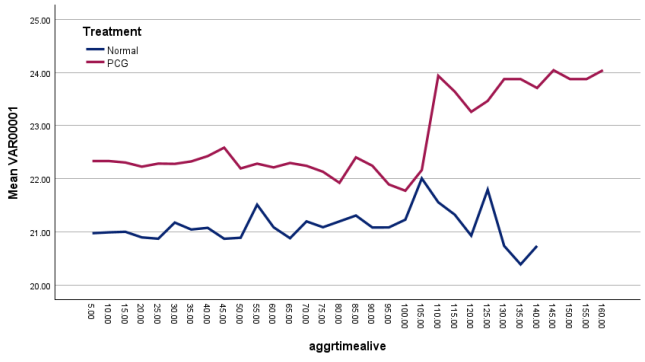


Figure 12: *Frustration over time for the 3 PCG and 1 normal games (average for all players).*

Another finding from exploring the results is that the frustration component does not fade over time. The graph above shows the total frustration averaged for all players and over time, separately for the 3 PCG games and the single normal game. Frustration is again averaged for each 5-second interval in-game to represent the time axis (X-axis). On the contrary, frustration is steeply increasing after the 100-second mark for PCG levels. This is not so apparent for normal games. Once again this chart reflects clearly the key finding of the ANOVA in favour of the significant difference of frustration between PCG and normal levels.

The above findings indicate that the unpredictable modification by the PCG algorithm is enough to keep frustration levels to a higher average, where it does not incite aggressive responses by the player. This can be attributed to the feeling that each PCG level is distinct and keeps things fresh and fun for the players. Also, in the chart we can see that PCG games, on average, take longer to complete, possibly due to the fact that the players do not have enough time to familiarize themselves with the behavioral changes and the changes in the level layout. On the other hand, it can be attributed of course to the increased difficulty provided by the PCG engine.

Finally, this also answers our **third research question**, namely that players generally do not get used to the randomization components, however, future studies could more accurately investigate this subject over much longer periods of time by administering more PCG and normal games.

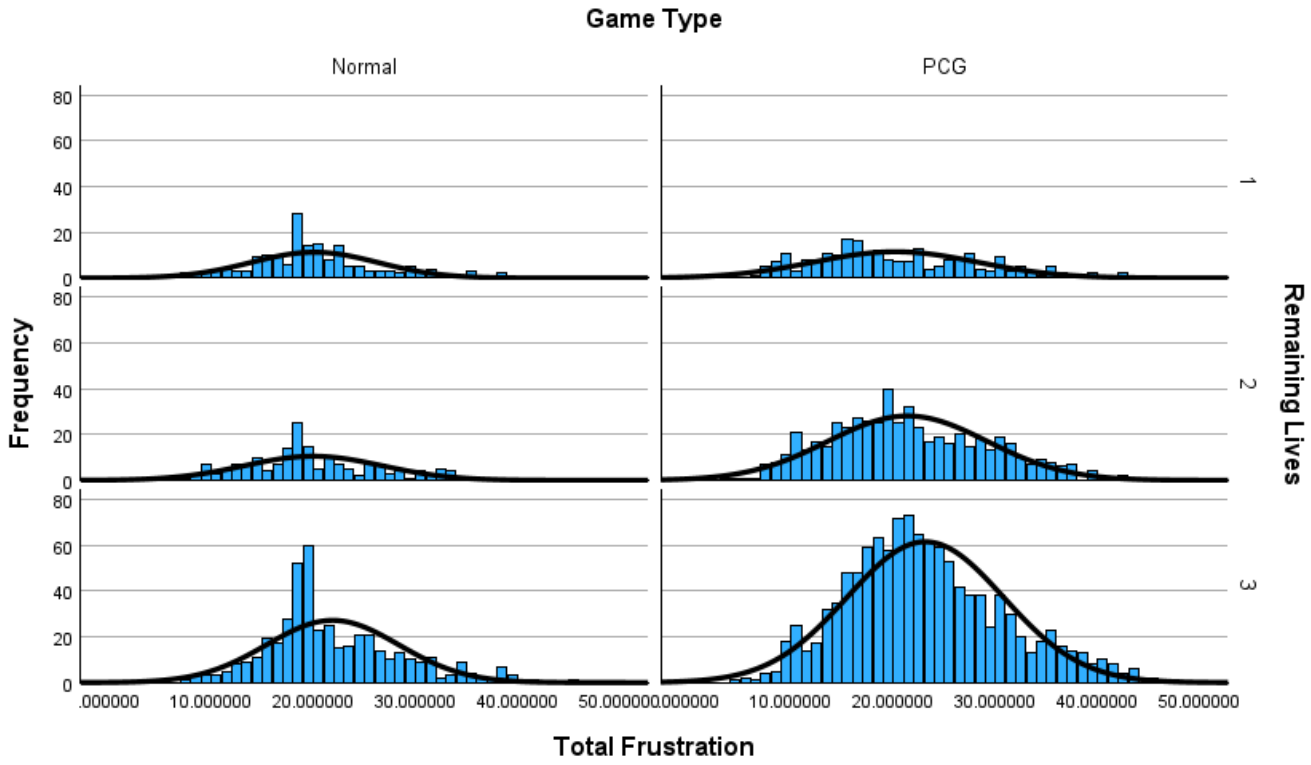


Figure 11: *Frequency plot of frustration over remaining lives, split by game type.*

A survival analysis approach to game completion

Finally there is another way to see how PCG can affect game duration. The below "life curve" shows that with PCG, survivability drops significantly at times between 50 and 100 seconds. We explain this due to the fact that frustration increases in PCG, resulting in some of the players having aggressive responses and losing lives earlier than in non-PCG levels.

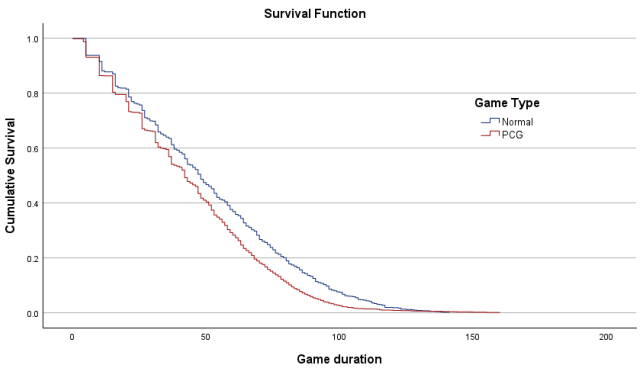


Figure 13: *Game duration - survival rate over time*

Limitations

The study acknowledges several limitations, such as the specific focus on Pac-Man and the use of self-reported measures. These limitations present opportunities for future research. Future studies could explore the effects of randomization across different game genres and employ objective measures (e.g., physiological or behavioral indicators) in addition to self-reported measures to provide a more comprehensive understanding of player experiences.

More specifically, eye-tracking and pulse measuring could add significantly richer layers of analysis and interpretation of various PCG tweaks.

FUTURE WORK

There are plenty of avenues for future work in this area of research. This paper shows that limiting the player's freedom, when done carefully, can induce cognitive and focus-inducing frustration, leading to an induced flow state. However, failure to apply a limiting factor in this behavior can lead to aggressive responses and feelings of failure. Future research could address this problem and find this limit.

There is always the prospect of applying the same rules mentioned here regarding limiting the player's freedom in the 3D space, where there isn't a sufficient amount of research exploring PCG and frustration. The current study focuses on the Pac-Man game, but future research could explore the effects of randomization on frustration and boredom across multiple game genres. Comparing different game genres (e.g., first-person shooters, puzzle games) would provide a more comprehensive understanding of how randomization influences player experiences across different types of games. Indeed, the most

important factor that could influence boredom and frustration in players is the sentimental detachment with the specific game and this factor unfortunately cannot be accounted for.

Also, the paper focuses on the immediate effects of randomization on frustration and boredom in games. A future study could investigate the long-term effects of randomization by examining how players' experiences change over multiple gameplay sessions. This would provide insights into whether players' frustration levels decrease or stabilize over time as they become more accustomed to randomization.

CONCLUSION

In this paper, an algorithm that modifies Pac-Man levels randomly through the use of noise-based PCG was described. A study was then conducted in which a number of participants played four games of Pac-Man, one without PCG and three with PCG randomization. After each game, the participants were asked about their experience in relation to the controls of the game and some components of frustration as mentioned in the frustration-aggression hypothesis, in conjunction with the flow theory.

The results revealed several interesting findings. Firstly, it was found that the randomization of the level layout and gameplay elements had a significant impact on player frustration. Participants reported higher levels of frustration in the levels with randomization compared to the non-randomized level. This suggests that the introduction of PCG can increase frustration in players.

Additionally, it was observed that the unfamiliarity with the level layout caused by randomization had a substantial effect on frustration. Participants experienced higher frustration when navigating through unfamiliar layouts compared to the familiar non-randomized layout. This finding indicates that the unpredictability introduced by PCG can contribute to frustration in players.

Interestingly, it was found that participants did not become significantly more accustomed to PCG towards the end of the gameplay session. This suggests that the frustration caused by randomization persisted throughout the experiment, indicating that players did not adapt to the unpredictable nature of PCG.

REFERENCES

[1] Leonard Berkowitz. 1978. Whatever Happened to the Frustration-Aggression Hypothesis? *American Behavioral Scientist* 21 (1978), 691–708. Issue 5. DOI: http://dx.doi.org/10.1177/000276427802100505

[2] Leonard Berkowitz. 1989. Frustration-Aggression Hypothesis: Examination and Reformulation. (1989). Issue 1.

[3] S. Cooper, A. El Rhalibi, Madjid Merabti, and J. Wetherall. 2010. Procedural content generation and level design for computer games. (01 2010), 35–40.

[4] Mihaly Csikszentmihalyi. 1990. *Flow: The Psychology of Optimal Experience*.

[5] Mihaly Csikszentmihalyi, Sami Abuhamdeh, and Jeanne Nakamura. 2005. *Flow.* Guilford Publications. 598–608 pages.

[6] John Dollard, Neal E Miller, Leonard W Doob, O H Mowrer, and Robert R Sears. 1939. *Frustration and aggression.* Yale University Press. 213, viii, 213–viii pages. DOI:http://dx.doi.org/10.1037/10022-000

[7] Dagmara Dziedzic. 2016. Dynamic difficulty adjustment systems for various game genres. *Homo Ludens* 1 (12 2016), 35–51.

[8] Mark Hendrikx, Sebastiaan Meijer, Joeri Velden, and Alexandru Iosup. 2013. Procedural Content Generation for Games: A Survey. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)* 9 (3 2013). DOI: http://dx.doi.org/10.1145/2422956.2422957

[9] Robin Hunicke. 2005. The case for dynamic difficulty adjustment in games. *ACM International Conference Proceeding Series* 265 (06 2005), 429–433. DOI: http://dx.doi.org/10.1145/1178477.1178573

[10] Martin Jennings-Teats, Gillian Smith, and Noah Wardrip-Fruin. 2010. Polymorph: Dynamic difficulty adjustment through level generation. (06 2010). DOI: http://dx.doi.org/10.1145/1814256.1814267

[11] Bo-Gyun Jeong, Sung Cho, and Shin Kang. 2014. Procedural Quest Generation by NPC in MMORPG. *Journal of Korea Game Society* 14 (02 2014). DOI: http://dx.doi.org/10.7583/JKGS.2014.14.1.19

[12] Seung A.Annie Jin. 2011. "I feel present. therefore, i experience flow:" A structural equation modeling approach to flow and presence in video games. *Journal of Broadcasting and Electronic Media* 55 (1 2011), 114–136. Issue 1. DOI: http://dx.doi.org/10.1080/08838151.2011.546248

[13] Olana Missura. 2015. *Dynamic Difficulty Adjustment*. Ph.D. Dissertation.

[14] Chris Pedersen, Julian Togelius, and Georgios N. Yannakakis. 2009. Modeling Player Experience in Super Mario Bros. In *Proceedings of the 5th International Conference on Computational Intelligence and Games (CIG'09)*. IEEE Press, 132–139.

[15] Jamey Pittman. 2009. The pac-man dossier. (Feb 2009). https://www.gamedeveloper.com/design/the-pac-man-dossier

[16] Gabriel Sepulveda, Felipe Besoain, and Nicolas A. Barriga. 2019. Exploring Dynamic Difficulty Adjustment in Videogames. 1–6. DOI: http://dx.doi.org/10.1109/CHILECON47746.2019.8988068

[17] Noor Shaker, Julian Togelius, and Mark Nelson. 2016. *Procedural Content Generation in Games*. DOI: http://dx.doi.org/10.1007/978-3-319-42716-4

[18] Mirna Silva, Victor Silva, and Luiz Chaimowicz. 2016. Dynamic Difficulty Adjustment on MOBA Games. *Entertainment Computing* 18 (10 2016). DOI: http://dx.doi.org/10.1016/j.entcom.2016.10.002

[19] Adi Stein, Yair Yotam, Rami Puzis, Guy Shani, and Meirav Taieb-Maimon. 2017. EEG-Triggered Dynamic Difficulty Adjustment for Multiplayer Games. *Entertainment Computing* 25 (12 2017). DOI: http://dx.doi.org/10.1016/j.entcom.2017.11.003

[20] Rhio Sutoyo, Davies Winata, Katherine Oliviani, and Dedy Martadinata. 2015. Dynamic Difficulty Adjustment in Tower Defence. *Procedia Computer Science* 59 (12 2015), 435–444. DOI: http://dx.doi.org/10.1016/j.procs.2015.07.563

[21] Roland van der Linden, Ricardo Lopes, and Rafael Bidarra. 2014. Procedural Generation of Dungeons. *IEEE Transactions on Computational Intelligence and AI in Games* 6 (2014), 78–89. Issue 1. DOI: http://dx.doi.org/10.1109/TCIAIG.2013.2290371

[22] Max Wolterink and Sander Bakkes. 2021. Towards explainable prediction of player frustration in video games. 1–10. DOI: http://dx.doi.org/10.1145/3472538.3472566