Towards Adapting a Content Orchestrator to a Different Game Genre: Generating Levels, Rules, and Narrative for Diverse Player Profiles from a Top-down Adventure to a 2D Platformer

Tyago Yuji Teoi[1],Leonardo Tórtoro Pereira[2]
Claudio Fabiano Motta Toledo[1]

[1] Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo (USP) - São Carlos, SP - Brasil

[2]Instituto de Geociências e Ciências Exatas
Universidade Estadual Paulista (UNESP) - Rio Claro, SP - Brasil

yujiteoi@gmail.com,leonardo.t.pereira@unesp.br,claudio@icmc.usp.br

Abstract. *Procedural Content Generation for multiple game content facets is a challenge for the game industry and academia. A content orchestrator is a software that can manage different procedural content generators, mixing their outputs while maintaining coherence and feasibility. We adapted a content orchestrator, originally meant for a top-down adventure game, to a 2D platformer. Both versions procedurally generate Levels, Rules, and Narrative, while adapting to distinct player profiles. A pre-test questionnaire is used to evaluate the player profile, and a post-test questionnaire is used to evaluate the game prototype we developed for this experimental purpose and the procedurally generated content. Results show the game was fun, challenging, interesting to explore, and with a moderate difficulty. Although with a limited sample, our results indicate the system was able to target content based on profiles. Therefore, this is a first step into understanding how a content orchestrator can be adapted to different game genres.*
Keywords *Procedural Content Generation, Content Orchestration, Game Levels, Game Narrative, Game Rules, Digital Games, Platformer*

1. Introduction

In the context of digital games, elements of uncertainty are fundamental for engaging player experience [Salen e Zimmerman 2003]. As in unbiased dice rolls or the rotation of a casino roulette wheel, these randomness factors play an important role in unpredictability. Games with predictable patterns can become monotonous [Koster 2005]. Furthermore, there are numerous motivations for players to play games. As described by Nick Yee [Yee 2006], different players can attribute different meanings to their gaming experiences.

Considering both randomness and content adaptation, Procedural Content Generation (PCG), defined as the algorithmic creation of game content, has been demonstrated to be an approach to generating novel and less repetitive personalized content [Shaker et al. 2016].

However, implementing PCG across multiple creative game facets (e.g., Rules, Visuals, and Levels) in the same game has been challenging for the game industry and academia, as coordinating these different content presents technical complexities [Liapis et al. 2019]. Thus, Liaps et al. propose a Content Orchestrator, aiming to achieve both diversity and coherence between different contents [Liapis et al. 2019].

The *Overlord* content orchestrator, presented in [Pereira et al. 2022], was able to orchestrate Levels, Rules (as enemy behavior), and Narrative (as quests) and adapt the content to different player profiles for a 2D top-down action-adventure game. We adapted the system to generate content for a 2D platformer game, trying to change the orchestrator and its content generators as little as possible.

A game prototype for the 2D platformer was created and shared over social networks and communities for users to play and send anonymous data. A pre-test questionnaire was used to identify the players' profiles, and a post-test questionnaire was used to evaluate each player's feedback on the game prototype and the content generated. Results show that, on average, most of the players found the game fun, liked the exploration and the challenge of key-lock puzzles, and found the difficulty average. Those who played a dungeon adapted to their profile found it more difficult, which may be due to most respondents preferring more difficult games.

Therefore, we adapted an existing content orchestrator to another (albeit similar) game genre with few changes. As far as we know, this is the first time such an adaptation between genres has been done in the literature for a content orchestrator. We hope our results pave the way for others to reuse existing orchestrators and create new ones that are flexible and extensible.

Next, we define the concepts of content orchestration (Section 1.1) and player profiling (Section 1.2), as they are essential to understanding our research. Then, we present our research questions and the hypothesis in Section 1.3.

1.1. Content Orchestration

The concept of Content Orchestration in the context of PCG was described by Liapis et al. [Liapis et al. 2019]. They define orchestration as the act of starting and maintaining the coherence of content generators across two or more creative facets (delimited by the author as Levels, Rules, Narrative, Audio, Visuals, and Gameplay).

The Orchestrator is the software responsible for the orchestration. The authors define three types of Orchestrators:

- **Top-down**: As in a symphony, the Orchestrator is the maestro, and the content generators are the musicians. The maestro makes the musicians play exactly as he commands;
- **Bottom-up**: As in a jazz band, the musicians (content generators) are free to improvise. The Orchestrator function is the same as the rhythm or base melody;
- **In-between**: This is a middle term for the above approaches.

1.2. Player Profiling

To adapt content to different players, we use the definition of four motivation factors, presented by Yee [Yee e Ducheneaut 2018]:

- **A**chievement: Players aiming to maximize in-game achievements or strengthen their characters.
- **C**reativity: Players who wish to personalize their characters, express themselves in the game, and explore creativity.
- **I**mmersion: Players interested in incorporating their persona into the playable character or creating narratives.
- **M**astery: Players seeking challenges, complex problems, and games involving strategic thinking.

Each factor is assigned a unique value between 1 and 4, where 1 represents the minimum influence and 4 denotes the maximum. For instance, a profile emphasizing **Mastery** would receive a value of 4 for that factor, while the other factors would be assigned values of 3, 2, and 1. The player profile is sent by the Content Orchestrator to all content generators. This is the same representation used in [Pereira et al. 2022], and was chosen so we may directly compare our results to theirs.

1.3. Research Questions and Hypothesis

We hypothesize that it is possible to reuse an *In-between* Content Orchestrator from one game genre to another by only translating the outputs from content generators to the new play-space and mechanics when required, while maintaining the profile adaptation from the Overlord system, at least for similar game genres.

The below research questions guided our hypothesis and study:

RQ-1 Can a Content Orchestrator be adapted from one game genre to another? More specifically, can an orchestrator for a top-down adventure game be adapted for a 2D platformer? How?

RQ-2 Can a Content Orchestrator for a 2D platformer game provide fun, diverse, and challenging content?

RQ-3 Can a Content Orchestrator for a 2D platformer game adapt its contents to different player profiles?

Section 2 will present some related work that corroborates with our hypothesis and shows the state-of-the-art of the field. Next, Section 3 will present the game prototype, while Section 4 will present the orchestration process and its generators. The experimental procedure and the results gathered from them will be shown in Section 5 and 6, respectively. While our limitations are discussed in Section 7 and, finally, our main conclusions are discussed in Section 8.

Table 1. Related work comparison

Reference	Game genre	Facet orchestration	Player evaluation	Profile adaptation	Multigenre
[Hartsook et al. 2011]	RPG	Levels and Narrative	No	Yes	No
[Treanor et al. 2012]	Arcade games	Levels, Rules, Visuals, and Narrative	No	No	Yes
[Cook et al. 2021]	Platformer	Levels, Audio, Visuals, and Narrative	No	No	Yes
[Karavolos et al. 2021]	FPS	Levels and Rules	No	No	No
[Migkotzidis e Liapis 2022]		Levels and Rules	No	No	No
[Pereira et al. 2022]	Top-down	Levels, Rules, and Narrative	Yes	Yes	No
[Hojatoleslami et al. 2024]	FPS	Levels, Audio, Visuals, and Narrative	Yes	No	No
Our study	Platformer	Levels, Rules and Narrative	Yes	Yes	Yes

2. Related Work

Here we present some of the most relevant and closely related recent works covering content orchestration, and compare them to our approach, and how they contribute to our hypothesis. Table 1 summarizes the similarities and novelties of our study.

Liapis et al. [Liapis et al. 2019] presented a comprehensive review of studies exploring content orchestration across diverse creative facets [Browne e Maire 2010, Cook et al. 2013, Lopes et al. 2016, Barros et al. 2018, A.B. et al. 2016, Green et al. 2018, Cook e Colton 2021, Hoover et al. 2015, Hartsook et al. 2011, Treanor et al. 2012, Cook et al. 2021]. Some selected studies are covered. The first is Hartsook et al.'s *Game Forge*, a system designed to generate fully playable Computer Role-Playing Games (CRPGs) procedurally. Their approach incorporates preferences for gameplay styles, design aesthetics, and unique story structures authored by humans or computational systems. The system orchestrates Narrative and Levels, leveraging generated questlines to create cohesive levels aligned with the overarching story goals. As in our approach, they use a questionnaire to identify the player's profile and adapt the content to them [Hartsook et al. 2011].

Although *Game Forge* is tested only in a 2D Zelda-like RPG prototype, the authors argue that *Game Forge* could be used in other story-based games. However, they have not tested their system in different game genres or had players evaluate the content generated.

Treanor et al. presented *Game-O-Matic*, an authoring tool for creating arcade games. The tool uses a feed-forward pipeline to represent ideas through a concept map input system. This pipeline maps verbs to game mechanics, generates micro-rhetorics, and applies recipes for coherence. They demonstrate their system can generate content for diverse types of arcade-style games, but it was not adapted to different profiles or tested with players [Treanor et al. 2012].

Cook et al. introduced *ANGELINA*, an automated video game design system that can autonomously select multimedia content for game themes. From news articles, the system orchestrates Visuals, Audio, and Narrative for a platformer game, while generating Levels independently. Their study showcases the potential of automated content selection from other sources for game design and can be used to create games of multiple genres[1]. But, as *Game-O-Matic*, the system has no profile adaptation nor was tested by players. Still, their studies show that Content Orchestration for platformer games is feasible [Cook et al. 2021].

Other researchers conducted novel studies after the review by Liapis et al. [Liapis et al. 2019], we present some of them.

Karavolos et al.'s framework uses a surrogate model to orchestrate the generation of Levels and Rules facets. The model extracts associations between these facets through deep learning on gameplay logs. The authors tested the framework in a one-versus-one First-Person Shooter (FPS). Results indicate that a balanced matchup and desired designs are more efficient when orchestrating classes and levels. The system was not tested with players [Karavolos et al. 2021].

Migkotzidis and Liapis present SuSketch, a mixed-initiative design tool to

[1] https://gamesbyangelina.itch.io/

generate and orchestrate Levels and Rules for FPS games [Migkotzidis e Liapis 2022]. Their study is an extension of Karavolos et al.'s framework, which uses surrogate models to suggest levels to the game designer through a user interface (UI) and presents additional gameplay metrics [Karavolos et al. 2021]. Game designers evaluated the tool's usability by performing specific tasks and answering a Post-Study System Usability Questionnaire (PSSUQ). Our work does not test the system's usability, though it tests the content generated by the orchestrator.

Pereira et al. introduced Overlord, a system for orchestrating PCG across three creative facets: Levels, Rules, and Narrative. Overlord adapts content based on player profiles, considering motivation factors such as Achievement, Creativity, Mastery, and Immersion. They tested the system in a game prototype from the top-down adventure genre, and validated it by having players answer questionnaires. Their results indicated that the system provided entertaining and challenging content [Pereira et al. 2022]. As a genre similar to a 2D platformer, with the source code public on GitHub, and presenting positive results, we decided to adapt their system. The main differences between our work and theirs are shown in Section 4.

Hojatoleslami et al. presented a General Framework for Generating Dungeons with Atmosphere (GFGDA) [Hojatoleslami et al. 2024]. The authors advocate that the correct arrangement of game elements creates the game atmosphere, and propose the CAGE pattern, based on player motivations of play [Schell 2008] to guide the order in which to present these elements to the player. The evaluation of atmosphere generation used a Game Experience Questionnaire (GEQ), presented after playing an FPS game. Their method of influencing the players' emotions showed positive results. Besides using different player motivations to play and generating joyful and eerie dungeons, their study does not adapt their content to different player profiles. Furthermore, the authors tested the system in only one game genre (i.e., FPS).

3. 2D Platformer Game Prototype

The 2D platformer game prototype consists of a playable character, a dungeon having rooms with one or two doors, enemies, collectibles, lore items, Non-Playable Characters (NPCs), keys, and a Sierpiński triangle (endgame item, refered from now on as Triangle). The objective of the game is to find the latter. On the way to find it, in addition to enemies, the player encounters locked doors. The player needs keys, collected through the dungeon, to unlock them.

However, during the game, the player can encounter one of the three NPCs and complete their quests, which provide rewards in collectibles or status-enhancing equipment.

The player's mechanics are limited to walking, jumping, and shooting arrows horizontally. The player starts with 20 hearts of life and dies when it comes to zero hearts. There is a minimap to help the players locate themselves in the dungeon.

For exemplification, Figure 1 shows a part of a dungeon room, containing the player, his current life, items collected at the top left, a minimap at the bottom right, an enemy, and an NPC. Figure 2 shows an example of a full dungeon. The purple squares behind the rooms are the view of the player's minimap.

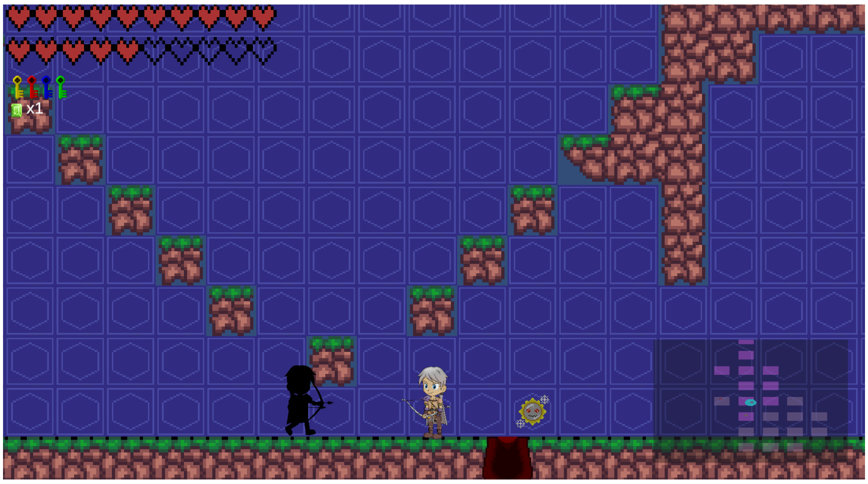


Figure 1. The game's screenshot. The player is on the bottom-center of the image, an enemy is on his left (the shadow figure), and an NPC is on his right side (the cog-like sprite). The bottom right has a mini-map of the dungeon, and the top left shows the player's health and collected items.

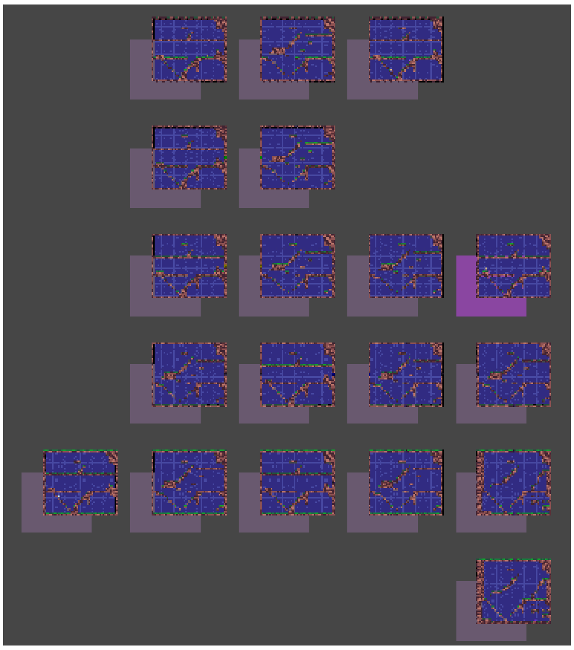


Figure 2. A dungeon sample generated by the Orchestrator. The blue squares represent the room background, and the purple squares form the mini-map.

The Platformer game and the Overlord orchestrator were developed using the Unity 3D Game Engine, using the C# language. Google's Firebase database was used to collect data, preserving the respondent's anonymity. The repository for both is available at *GitHub*[2].

4. Content Orchestration

In this section, we outline changes made to [Pereira et al. 2022]'s Overlord system to make it orchestrate content for a 2D Platformer game. We also describe the game elements related to the generated content.

4.1. Levels

Our game's Levels facet is represented by the dungeon, its rooms, and the spawn position of enemies, NPCs, and items (including collectibles, lore items, equipment, and the Triangle) in said rooms.

Considering a platformer game compared to a top-down game, generally, the former has gravity mechanics whereas the latter does not, with major implications for gameplay. Consequently, we need to guarantee reachable spawn-point positions for game objects like items, enemies, and NPCs. Additionally, enemies and NPCs need to spawn on a solid block (since they do not fly in our game). Furthermore, we cannot generate rooms with locked spaces or places the player cannot reach (e.g., too high).

To address these problems, a backtracking algorithm was developed. It calculates the spawn points for items, enemies, and NPCs on top of a room's walkable block. Figure 3, shows a generated room model, and the yellow points indicate all possible spawn locations calculated by the algorithm, presented in Algorithm 1.

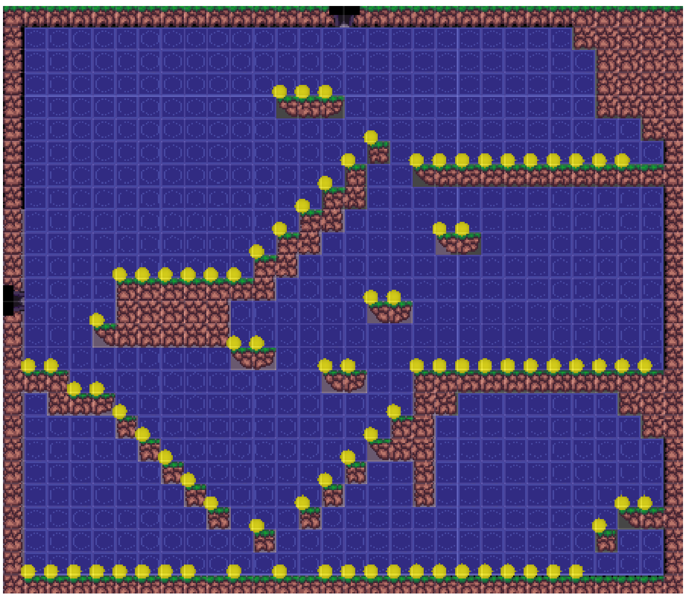


Figure 3. A room. Blue tiles are the background, and brown ones are the ground blocks. The yellow circles are spawn points.

[2] https://github.com/LeonardoTPereira/Overlord-Project/tree/2d-game/feature/enemy-movements

```
Data: Position position, PathMatrix path[][], SpawnPointList spawnPoints[]
if position is a ground block then
    return
if bellowposition is a ground block and player can stand then
    path[bellowposition.y[bellowposition.x] ← True
    spawnPoints.Add(position)
// In our case nextposition can be the UP, DOWN, LEFT,
or RIGHT block from position
nextPositions[] ← CalculateNextPositions(position)
i ← 0
while i ¡ nextPositions.length do
    if path[nextPositions[i].y][nextPositions[i].x] equal to False then
        path[nextPositions[i].y][nextPositions[i].x] ← True
        CalculateSpawnPoint(nextPositions[i], path, spawnPoints)
    i ← i + 1
```

Algorithm 1: Pseudocode for the backtracking algorithm. Tracks all positions above a reachable block in a room, considering the player's character height and jump physics.

The dungeon generation is achieved using the same Evolutionary Algorithm presented in [Pereira et al. 2022], and the room generation was altered to randomly select one of many pre-defined room templates. This modification was needed since their room-generation algorithm did not take gravity into account.

4.2. Rules

In our work, Rules are represented as enemies' parameters. We used a MAP-Elites approach to generate both diverse and optimal enemies, the same as in [Pereira et al. 2022] and better detailed in [Viana et al. 2022].

For our game, we created five enemies and seven movements. Each enemy has a parameter value for health, damage, projectile speed, attack speed, movement speed, active time, and rest time. Figure 4 shows all possible enemies and their movements.

We did not alter any of the parameters from [Pereira et al. 2022] and [Viana et al. 2022], except by finding new minimum and maximum values for the parameters above. We tested and fine-tuned the parameters until we felt that their predefined fitness values for easy, medium, and hard difficulties were reflected in the game. New values were needed as the enemies and gameplay were different between genres. Furthermore, the scripts for the movement and attack behaviors for each enemy had to be altered to reflect a 2D platformer's gameplay.

4.3. Narrative

The Narrative facet in our game is structured around *quests*, introduced when the player interacts with one of the three NPCs around the dungeon. The Quest Generator selects specific types of missions based on the player's profile, and they guide other generators, as in [Pereira et al. 2022]. These missions can fall into the categories:

- Achievement: Exchange collectible items for equipment or collect a certain quantity of a specific item.

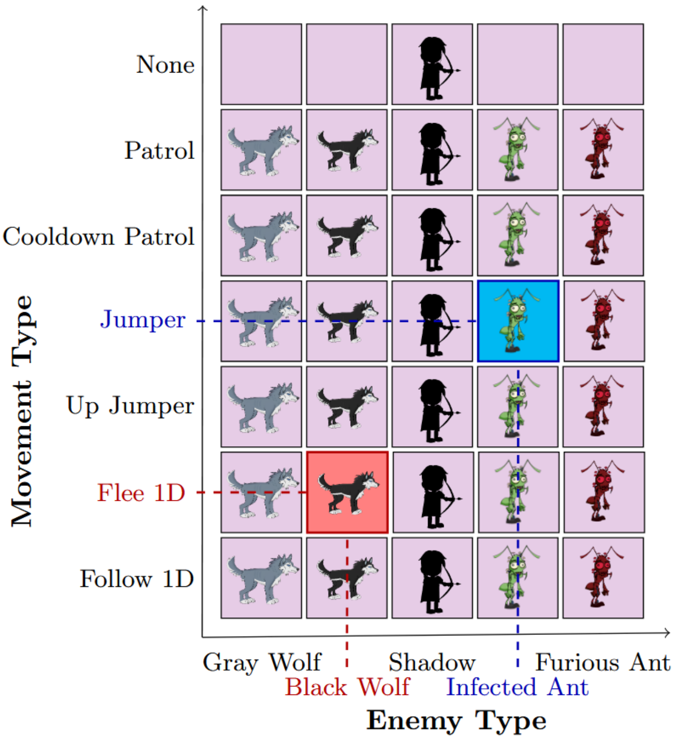


Figure 4. MAP-Elites' map of enemies and movements. The blue square is an Infected Ant with Jumper movement, and the red square is a Black Wolf with Flee 1D movement.

- Creativity: Explore a specific room or visit a determined number of rooms in the dungeon.
- Immersion: Communicate with or give an item to an NPC, read an item of *lore*[3].
- Mastery: Defeat or cause a certain amount of damage to a designated enemy.

The quest generation uses the Formal Grammar $G = (N, \Sigma, P, S)$, adapted from [Pereira et al. 2022] to contain more quests. The non-terminal symbols are $N = S, A, C, I, M$, where each, except the initial symbol S, represents a mission type described above, corresponding to the initial letter of each mission type.

The terminal symbols are $\Sigma = \{\varepsilon$, explore, go-to, collect, exchange, defeat, listen, read, deliver, report$\}$. They respectively represent an empty quest, visit a determined number of rooms, visit a specific room, collect a determined number of items, exchange an item with an NPC, defeat a specific enemy, talk to an NPC, read an item of *lore*, deliver a specific item to another NPC, and talk to another specific NPC. The production rules P are shown below:

- $S \rightarrow C|A|M|I$
- $C \rightarrow$ explore $C|$go-to $C|\varepsilon$
- $A \rightarrow$ collect $A|$exchange $A|\varepsilon$
- $M \rightarrow$ defeat $M|\varepsilon$
- $I \rightarrow$ listen $I|$read $I|$deliver $I|$report $I|\varepsilon$

[3]In this game, *lore* items are interactable items generated in the maps that convey parts of the game's story.

Once generated, the missions are assigned to NPCs and presented to the player through dialogue. Each NPC possesses a name and a profession, imbuing authenticity and spontaneity to dialogues. All dialogues were human-produced, with keywords in the dialogues replaced by the quest's specification given by the Quest Generator.

The dialogue system is the same as described by Pereira et al. [Pereira et al. 2022], but translated into Portuguese, with new names of items and enemies to reflect the characteristics of our game.

4.4. Orchestrator

We used the orchestrator which is an In-between approach [Pereira et al. 2022]. The Levels, Rules, and Narrative generators are started by the Orchestrator and the player profile is used as input. Moreover, it is possible to set a minimum and maximum value for the number of rooms, enemies, and items.

Firstly, as shown in Figure 5, the orchestration involves a Profile Analyst, which determines the weight of motivation factors, defined in Section 5. For this study, we used a pre-test questionnaire as input for the Profile Analyst, but the orchestrator can also calculate the profile weights from the post-test questionnaire.

Then, the profile weights are used by the Narrative generator to create quests. Subsequently, the quest data and player profile are sent to the Levels and Rules generators. As quests are input for both generators, we guarantee all of them can be finished. Moreover, the orchestrator is responsible for post-processing the generated content following some design decisions. As an example, some created enemies may have a melee attack and the *None* movement, which is undesirable and, thus, removed.

Finally, after the player selects a desired dungeon, the data generated by the system are used to instantiate the complete dungeon.

5. Experimental Settings

A playable version of the Platform Game was created as an executable for the Windows 10 operating system. This build was shared across gaming and technology communities on *Discord*[4] and social networks. No personal data was collected to preserve anonymity.

In the game, players underwent a one-time pre-test questionnaire to assess their player profiles. It comprised four questions, each assessing a motivational factor. Responses are provided on a 5-point Likert scale, from 1 (strongly disagree) to 5 (strongly agree). The factors and associated statements are as follows:

- Achievement:
  - I enjoy playing games where I can collect rare items and hidden treasures.
  - I like completing all missions, including those not necessary to finish the game.
- Creativity:
  - I enjoy playing games where I can explore the game world and discover secrets and mysteries.
  - I like exploring places, elements, and characters in a virtual world.

[4] A free voice and text application that enables community creation.

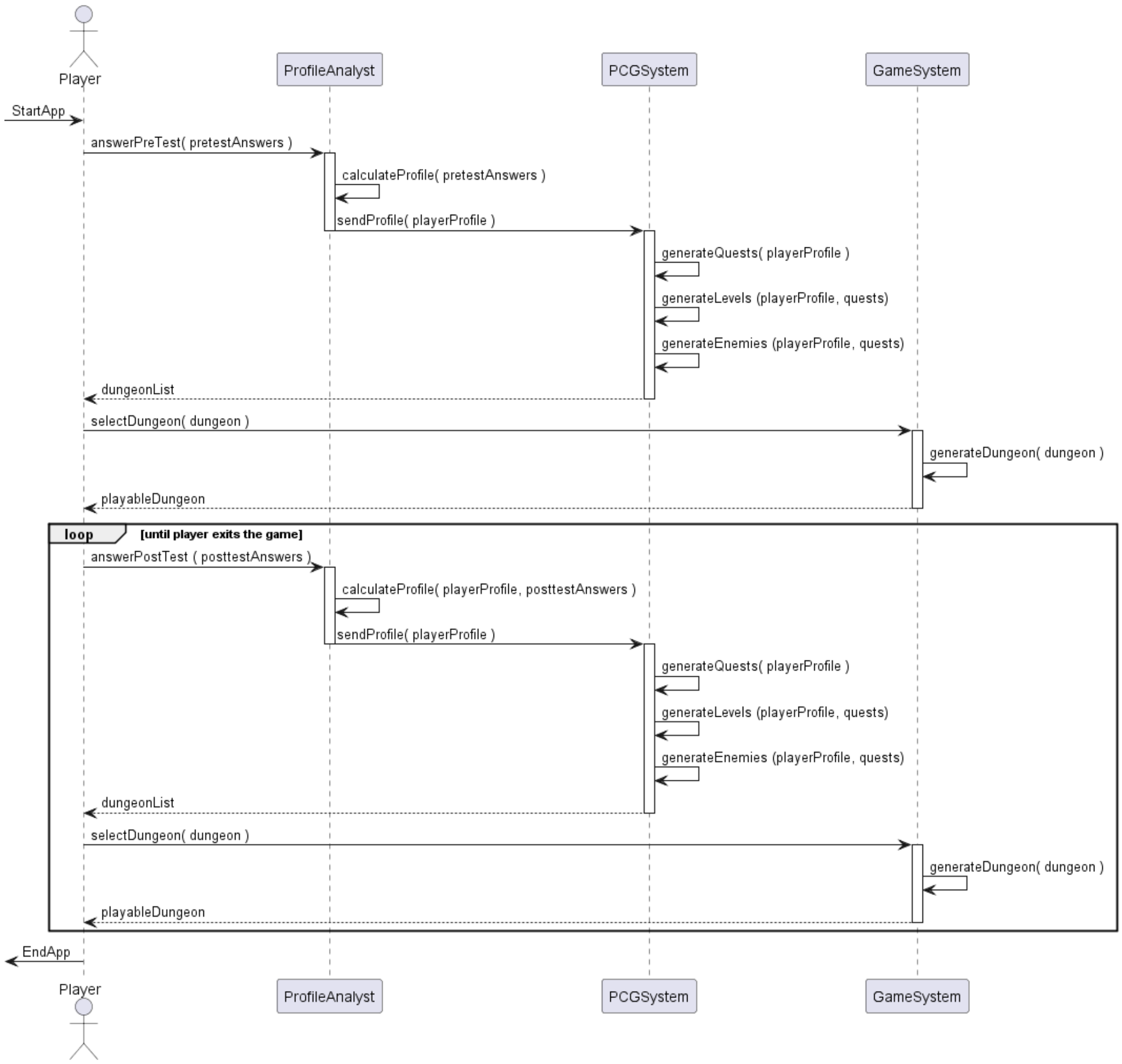


Figure 5. Sequence Diagram of the Orchestration Process. The player answers the pre-test questionnaire once, and the ProfileAnalyst uses the answers to calculate a player profile, which is sent to the PCGSystem to procedurally generate Quests, Levels, and Enemies. Then, the player selects a dungeon to play, and the GameSystem generates a playable dungeon. After playing the dungeon, the player answers a post-test questionnaire, which is used to adjust the player profile. The dungeon generation continues as described above until the player exits the game.

- Immersion:
  - I enjoy playing games where I can immerse myself in the character's role and make meaningful decisions.
  - I like forming friendships between game characters and working towards a common goal.
- Mastery:
  - I enjoy playing games where I can explode things, crush things, destroy things, shoot enemies, and kill enemies.
  - I like engaging in melee combat skills and dodging fast attacks.

After a dungeon is exited, either by finding the Triangle, giving up after losing all life points, or returning to the main menu through the pause menu, a post-test questionnaire is presented. It consists of 11 questions, answered by the same 5-point Likert scale:

- **Q1**: To what extent do you agree with the statement: "This level was fun to play"?
- **Q2**: To what extent do you agree with the statement: "This level was difficult to complete"?
- **Q3**: To what extent do you agree with the statement: "The enemies in this level were difficult to defeat"?
- **Q4**: To what extent do you agree with the statement: "The challenge of this level was just right"?
- **Q5**: To what extent do you agree with this statement: "The rewards of this level were just right"?
- **Q6**: To what extent do you agree with the statement: "I liked the amount of exploration available in this level"?
- **Q7**: To what extent do you agree with the statement: "I liked the challenge provided in finding the keys in this level"?
- **Q8**: To what extent do you agree with the statement: "It was difficult to find the exit in this level"?
- **Q9**: How diverse did you find the enemies in the level?
- **Q10**: To what extent do you agree with the statement: "I had fun completing the missions"?
- **Q11**: To what extent do you agree with the statement: "I liked the challenge provided in completing the missions"?

After submitting the post-test questionnaire, the dungeon associated with that questionnaire is locked for the player, preventing them from selecting the same dungeon twice.

6. Results

In this section, we present the outcomes derived from both pre-test and post-test questionnaires, later applying statistical hypothesis analysis to validate the results from the post-test.

6.1. Pre-test

39 individuals answered the pre-test questionnaire. Figure 6 illustrates the distribution of profiles obtained. Some profiles had few players, like *A=2, C=1, I=4 M=3*, which had a single one, and *A=1, C=3, I=4, M=2*, with two players. However, four of them had five players, as in *A=2, C=1, I=3, M=4*, and the most common profile was *A=1, C=3, I=2, M=4*, with seven players. We note that the majority of players had a higher value for Mastery, that is, liked challenging combat, which may have impacted the post-test results.

6.2. Post-test

The 39 users played a total of 51 dungeons, with an average of 1.31 dungeons per person, a variance of 0.42, and a standard deviation of 0.65. Of the 51 attempted dungeons, in 2 of them the players did not answer the post-test questionnaire, discarding them. From

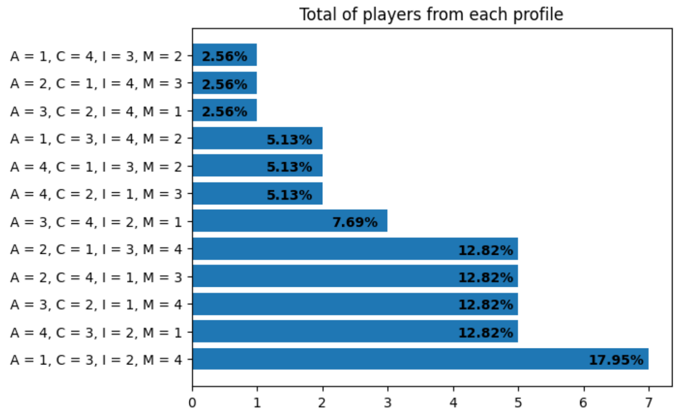


Figure 6. Distribution of profiles according to motivation factors: Achivement, Creativity, Immersion, and Mastery. Values range from 1 to 4.

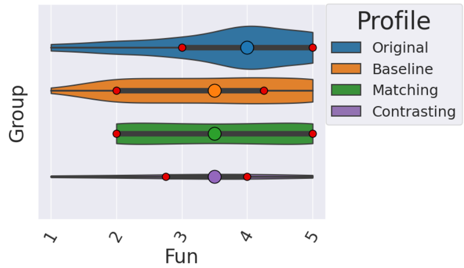
the remaining 49, if the player did not answer a specific post-test question, it was also not considered for the following results.

To test if orchestrating content matching the player's profile would enhance their gameplay, we divided users into two groups: the *Matching* one, where the dungeon had the same profile as the player (the test group), and the *Contrasting* one (the control group), the inverse case. This approach was used, instead of [Pereira et al. 2022]'s discrimination of players into profile, as we had fewer data. Our gameplay data is still skewed, with only 9 attempts for the *Contrasting* group, and 39 for the *Matching* one. To help compare the results, we also show the *Baseline*, which is the data considering both groups, and *Original*, the data from [Pereira et al. 2022], used with the authors' approval.

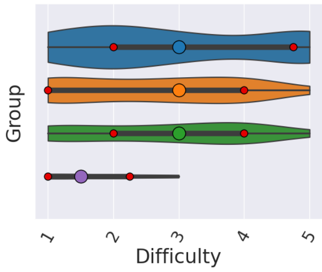
The graphs in Figure 7 display the responses, by player profile. For each violin graph reported, the four aforementioned groups are compared. The median is highlighted as the largest circle, and the first and third quartiles are highlighted as the smaller red circles. The width of each plot represents the number of respondents.

We observe some tendencies over the graphs. For the *Baseline*, some aspects were considered positive (median above 3), such as the fun, exploration, and the challenge of opening the locks (figures 7a, 7f, 7g, respectively). That is, the game is fun, and the exploration aspects are well-received. Others had a neutral opinion (median 3 and quartiles 2 and 4): challenge, difficulty finding the exit, and the fun and challenge of the quests (figures 7d,7h,7j, 7k, respectively). Meaning that the challenge (overall and for quests) and finding the exit were perceived as balanced, the quests were neutral.

The challenge of opening locks being positive, and the difficulty of doing so being neutral corroborates with themselves, as players tend to find balanced difficulties as better challenges, matching the Flow theory [Csikszentmihalyi e Csikzentmihaly 1990]. This is reflected in the overall challenge, as the difficulty and enemy difficulty were neutral, but leaning more to the smaller side, as the first quartile equals 1 (figures 7b and 7c, respectively). Meaning that, as the enemies were perceived as not very difficult, their challenge was perceived as more neutral. Finally, the rewards and enemy diversity were considered a little negative, with median below 3 (figures 7e,7i). This means our game needs to have more enemies implemented (as well as the MAP-Elites generator in Section

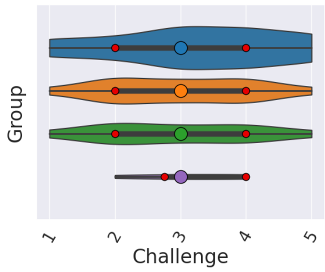


(a) How much fun each group felt.

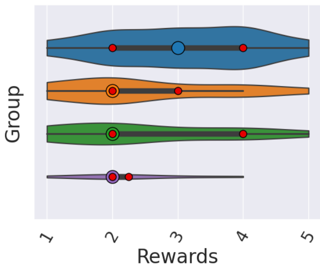


(b) Difficulty each group felt.

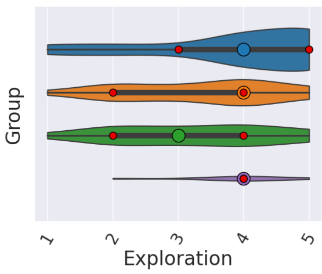
(c) Enemy difficulty each group felt.



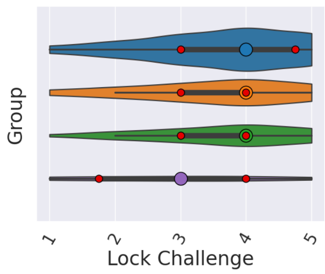
(d) How each group liked the challenge.



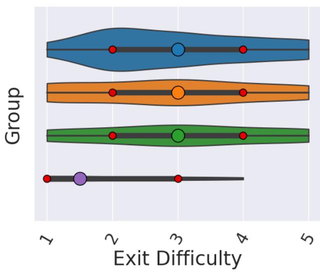
(e) How each group liked the rewards.



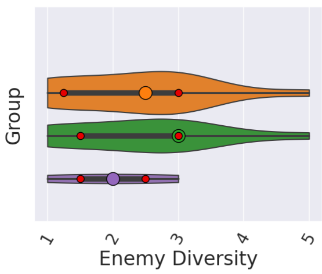
(f) How each group liked the exploration.



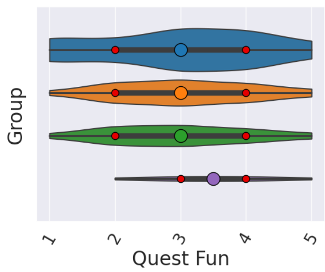
(g) How was the challenge of finding keys.



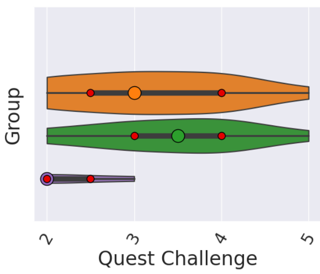
(h) How difficult was finding the exit.



(i) How much diversity the enemies had.



(j) How much fun was doing the quests.



(k) How challenging the quests were.

Figure 7. Violin plot comparing the post-test answers for each question divided by groups: Baseline (all answers), Matching (player and dungeon profile matched), and Contrasting (player and dungeon profile contrasting).

4.2), and a better reward system for players.

Finally, when compared to the *Original* data, we observe a strong similarity in player opinion compared to the *Baseline*. The differences were that the original game was considered a little more fun, difficult, having better rewards, more challenging locks to open, and more fun to explore. This may be due to top-down adventure games being more suited to exploration, or because the combat was a little harder and, thus, more rewarding. Either way, this similarity shows that our adaptation maintained the main characteristics of the original orchestrator when used in a different game genre.

6.2.1. Matching Profiles

Now, we evaluate if guiding the content generation towards player profiles affected player's opinions. For some questions, it appears that the *Contrasting* group had a more positive answer than the *Matching* or the *Baseline*, such as about fun (Figure 7a), challenge (Figure 7d), exploration (Figure 7f), and how much the quests were fun and challenging (figures 7j and 7k).

As this specific group had very limited responses, we cannot consider this tendency as a fact for the whole population. However, looking at this data holistically, the *Contrasting* group may have got somewhat easier dungeons. Specially considering most players had profiles leading to higher enemy difficulty (high Mastery trait). This could explain such higher evaluation.

Specially if we consider the questions where the *Matching* group had more positive answers than the *Baseline* and *Contrasting*: difficulty (Figure 7b), enemy difficulty (Figure 7c), rewards (Figure 7e), and enemy diversity (Figure 7i). The *Matching* group, holding players that, as seen in Figure 6, in majority preferred more challenging enemies, felt that they, in fact, faced harder enemies. This may have made the rewards seem more enjoyable, and made them face more enemies, increasing the feeling of diversity.

For the lock challenge (Figure 7g) and exit difficulty (Figure 7h), although the *Matching* group had a higher opinion over the *Contrasting* one, it had the same distribution as the *Baseline*. This means that there is little difference in the opinion of the groups, regarding the sample we observed.

6.3. Statistical Hypothesis Evaluation

To further validate if one of the observed trends in data may be significant, we conducted a statistical hypothesis analysis, considering the *Matching* and *Contrasting* groups. We compared the mean between groups using a Mann-Whitney U test, as the samples are not normalized. This was determined using a Shapiro-Wilk test applied to each sample to determine normality. These results are presented in Table 2.

For our null hypothesis, we considered a one-tailed test, selecting each tail according to the tendency observed in Figure 7. This allows us to reject the null hypothesis that the sample's mean is less than or equal to the population's mean when we want to confirm if a sample's mean is larger, and vice versa when we want to confirm if its mean is smaller than the population's mean. The tail, shown in Table 2, is presented as **G**(reater) if we are testing if the *Matching* population has a larger mean than *Contrasting*, or **L**(esser) for the opposite.

The one-tailed test provides a greater probability of rejecting the null hypothesis with the same alpha (0.05% in this study) compared to a two-tailed test, given that we know the tendency we are trying to confirm, proper as we have small samples.

Before presenting said analysis, we measured the power of our statistical hypothesis. As we have a small sample, we consider a Cohen's d (effect size) of 1.2 (very large), as presented in [Sawilowsky 2009]. By statistical power, we mean the probability of a hypothesis test to find an effect if it exists. It means the probability of committing a type II error, wrongly failing to reject the null hypothesis. A large effect size means that we can guarantee in most cases that results that do not reject the null hypothesis are correct if the difference between populations is very noticeable. Therefore, if our statistical hypothesis does not reject the null hypothesis but has a power above 0.8, a considerable difference in results is not noticeable, but a smaller one might be detectable with more data.

Table 2. The statistical hypothesis evaluation of post-test answers. The test considered that rejecting the null hypothesis means either that the *Matching* group's answer had a greater mean (Tail side = G) or lesser mean (Tail side = L) than the *Contrasting* group. All samples are not normal and had their power evaluated considering a very large Cohen's d effect side (1.2).

Question	Fun	Difficulty	Enemy Diff.	**Challenge**	Rewards	**Exploration**	Lock Chall.	Exit Diff.	Enemy Div.	**Quest Fun**	**Quest Chall.**
Power D=1.2	0.915	0.917	0.915	**0.000**	0.916	**0.00**	0.917	0.915	0.915	**0.00**	**0.00**
Normality p	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Hypothesis p	0.336	0.019	0.035	0.294	0.37	0.08	0.068	0.027	0.067	0.12	0.397
Tail side	G	G	G	L	G	L	G	G	G	L	L

Table 2 presents, for each post-test question, the results when comparing the *Matching* and *Contrasting* groups. The second row, *Power D=1.2*, shows the results for the test's power. For every test with the *Tail side* row containing *G* (indicating that rejecting the null hypothesis means the *Matching* group's mean is greater than the *Contrasting* group's mean), the power was above 90%. Namely, if the hypothesis is not rejected, concluding both groups share the same mean is unlikely to be an error. If we cannot reject the null hypothesis for the cases where *Tail side* equals *L* (highlighted in bold), as it has a power of 0, the same conclusion has a high chance of being an error. The result may change with more data.

The third row, *Normality p*, presents the p-values from the Shapiro-Wilk normality test for each sample, indicating that all samples are not normally distributed. The fourth row, *Hypothesis p* presents the p-values from the Mann-Whitney U test. For tests that could reject the null hypothesis with 95

From Table 2, we observe that for the Difficulty, Enemy Difficulty, and Exit Difficulty answers (names underscored), the *Matching* group provided significantly greater values than the *Contrasting* group. This suggests that when the dungeon matches the profile, players (mostly with high combat profiles) perceive the game as more difficult. However, for the other aspects, we cannot disregard that the answers may be the same and more data is needed to confirm, especially for the highlighted columns: *Challenge*, *Exploration*, *Quest Fun*, and *Quest Challenge*.

6.4. Answers to Research Questions

6.4.1. RQ-1: Can a Content Orchestrator be adapted from one game genre to another? More specifically, can an orchestrator for a top-down adventure game be adapted for a 2D platformer? How?

Yes, we can adapt it, at least to a close genre like from a top-down adventure to a 2D platformer. Section 4 presented the changes in the orchestration process. We discussed how the orchestrator did not need any changes, and the few changes done in each generator. In summary, we had to rebalance the minimum and maximum attributes for the enemy generator (and implement the new enemies mechanics and movements), alter the room generator to consider gravity and unreachable places caused by it.

Shared game rules (e.g., enemy life, speed, and damage) and the generation of dungeons and narrative could be reused, as well as the player profile adaptation. However, the rooms and spawn points were adjusted to permit the player to reach the enemies, items, NPCs, and room doors. Nonetheless, the whole orchestration process was presented for reproducibility. We also observed, through Figure 7's comparison of *Original* (answers from [Pereira et al. 2022]'s work) and *Baseline*, that the players' opinions were very similar. This indicates the translation was not only feasible via programming, but also able to provide content of similar quality to the original orchestrator.

6.4.2. RQ-2: Can a Content Orchestrator for a 2D platformer game provide fun, diverse, and challenging content?

As shown in Figure 7 and previously discussed, the *Baseline* answers show that the game was perceived as fun, although a little less than the *Original* answers, from [Pereira et al. 2022]. As previously mentioned, the answers were similar to their work, showing the adaptation was successful. Furthermore, users perceived the game as having a balanced difficulty and challenges, needing better rewards, enemy exploration and quest challenges, but having a good exploration and challenge to open locks.

6.4.3. RQ-3: Can a Content Orchestrator for a 2D platformer game adapt its contents to different player profiles?

As shown in Figure 7 and tested statistically in Table 2, the *Matching* users, when compared to *Contrasting* ones, found the game more difficult in all aspects. As the players were majorly oriented towards Mastery, which demands higher difficulty, we consider the profile-oriented content successful, as they perceived a higher difficulty. Yet, more balancing is needed, so they may consider the game more fun.

7. Limitations

Unfortunately, the sample size was not sufficient neither balanced enough to statistically validate all results. The *Matching* group had four times more playthroughs than the *Contrasting* one. Finally, for anonymity, the game did not collect any personal data. This allows a player to play the game on another computer or reset its pre-test data on the computer, which is considered as having a new player.

8. Conclusion

This study introduced a 2D Platformer game that adapted a Content Orchestrator from a top-down adventure one, from [Pereira et al. 2022]. This system was able to generate content for the facets of Levels (dungeons and rooms), Rules (enemy's behavior and attributes), and Narrative (quests), adapted for different player profiles. The translation process was discussed, showing that few changes were needed. Majorly on the room and enemy generation, but still simple changes.

The translation process was feasible and well-received. Results from Figure 7 indicate players' opinions (*Baseline*) were similar to the ones from the (*Original*) work. Moreover, the game was considered fun, challenging, and having a good exploration mechanic. However, more enemy diversity and better rewards are needed.

Yannakakis and Togelius [Yannakakis e Togelius 2018] state that game AI research has as frontier the development of general AI methods, encompassing game generality (methods applicable to any game), task generality (methods applicable to multiple related tasks), and user/designer/player generality (methods that adapt to different users, designers, or players). We attempt to advance these topics and outline the potential of a Content Orchestrator reusable to different game genres.

For future work, the orchestrator could explore the orchestration of more creative facets, such as Audio, Visuals, or Gameplay, enabling the procedural generation of scenery, enemy art, and associated sounds. Furthermore, a framework could be developed to facilitate its general use in industry or academia.

Acknowledgments

This work was funded by National Concil for Scientific and Technological Development - CNPq.

References

A.B., B. G., Antonios, L., e Julian, T. (2016). Playing with data: Procedural generation of adventures from open data. In *DiGRA/FDG &#3916 - Proceedings of the First International Joint Conference of DiGRA and FDG*, Dundee, Scotland. Digital Games Research Association and Society for the Advancement of the Science of Digital Games.

Barros, G., Green, M., Liapis, A., e Togelius, J. (2018). Who killed albert einstein? from open data to murder mystery games. *IEEE Transactions on Games*, PP.

Browne, C. e Maire, F. (2010). Evolutionary game design. *Computational Intelligence and AI in Games, IEEE Transactions on*, 2:1 – 16.

Cook, M. e Colton, S. (2021). A rogue dream: Automatically generating meaningful content for games. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 10(3):2–7.

Cook, M., Colton, S., e Pease, A. (2021). Aesthetic considerations for automated platformer design. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 8(1):124–129.

Cook, M., Colton, S., Raad, A., e Gow, J. (2013). Mechanic miner: Reflection-driven game mechanic discovery and level design. In Esparcia-Alcázar, A. I., editor, *Applications of Evolutionary Computation*, pages 284–293, Berlin, Heidelberg. Springer Berlin Heidelberg.

Csikszentmihalyi, M. e Csikzentmihaly, M. (1990). *Flow: The psychology of optimal experience*, volume 1990. Harper & Row New York.

Green, M. C., Barros, G. A. B., Liapis, A., e Togelius, J. (2018). Data agent. In *Proceedings of the 13th International Conference on the Foundations of Digital Games*, FDG '18, New York, NY, USA. Association for Computing Machinery.

Hartsook, K., Zook, A., Das, S., e Riedl, M. O. (2011). Toward supporting stories with procedurally generated game worlds. In *2011 IEEE Conference on Computational Intelligence and Games (CIG'11)*, pages 297–304.

Hojatoleslami, M. R., Zamanifar, K., e Zojaji, Z. (2024). Gfgda: general framework for generating dungeons with atmosphere. *Multimedia Tools and Applications*.

Hoover, A., Cachia, W., Liapis, A., e Yannakakis, G. (2015). Audioinspace: Exploring the creative fusion of generative audio, visuals and gameplay. In Carballal, A., Correia, J., e Johnson, C., editors, *Evolutionary and Biologically Inspired Music, Sound, Art and Design - 4th International Conference, EvoMUSART 2015, Proceedings*, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), pages 101–112, Germany. Springer Verlag.

Karavolos, D., Liapis, A., e Yannakakis, G. (2021). A multifaceted surrogate model for search-based procedural content generation. *IEEE Transactions on Games*, 13(1):11–22.

Koster, R. (2005). *A Theory of Fun for Game Design*. Paraglyph Press, Arizona.

Liapis, A., Yannakakis, G. N., Nelson, M. J., Preuss, M., e Bidarra, R. (2019). Orchestrating game generation. *IEEE Transactions on Games*, 11(1):48–68.

Lopes, P. L., Liapis, A., e Yannakakis, G. N. (2016). Framing tension for game generation. In *International Conference on Innovative Computing and Cloud Computing*.

Migkotzidis, P. e Liapis, A. (2022). Susketch: Surrogate models of gameplay as a design assistant. *IEEE Transactions on Games*, 14(2):273–283.

Pereira, L. T., Viana, B. M. F., e Toledo, C. F. M. (2022). A system for orchestrating multiple procedurally generated content for different player profiles. *IEEE Transactions on Games*, pages 1–11.

Salen, K. e Zimmerman, E. (2003). *Rules of Play: Game Design Fundamentals*. The MIT Press.

Sawilowsky, S. (2009). New effect size rules of thumb. *Journal of Modern Applied Statistical Methods*, 8:597–599.

Schell, J. (2008). *The Art of Game Design: A book of lenses*. Morgan Kaufmann.

Shaker, N., Togelius, J., e Nelson, M. J. (2016). *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*. Springer.

Treanor, M., Blackford, B., Mateas, M., e Bogost, I. (2012). Game-o-matic: Generating videogames that represent ideas. In *Proceedings of the The Third Workshop on Procedural Content Generation in Games*, PCG'12, page 1–8, New York, NY, USA. Association for Computing Machinery.

Viana, B. M. F., Pereira, L. T., e Toledo, C. F. M. (2022). Illuminating the space of enemies through map-elites.

Yannakakis, G. N. e Togelius, J. (2018). *Frontiers of Game AI Research*, pages 279–291. Springer International Publishing, Cham.

Yee, N. (2006). Motivations for play in online games. *CyberPsychology & Behavior*, 9(6):772–775. PMID: 17201605.

Yee, N. e Ducheneaut, N. (2018). 485Gamer motivation profiling: uses and applications. In *Games User Research*. Oxford University Press.