Proceedings of the Twentieth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE 2024)

# Generating Game Levels by Defining Player Experiences

**Pedro M. Fernandes, Manuel Lopes, Rui Prada**

INESC-ID and Instituto Superior Técnico, Univ. de Lisboa, Portugal
pedro.miguel.rocha.fernandes@tecnico.ulisboa.pt, manuel.lopes@tecnico.ulisboa.pt, rui.prada@tecnico.ulisboa.pt

### Abstract

This paper proposes a novel pipeline for generating game levels that elicit predefined emotional experiences from players. Our approach uses evolutionary algorithms alongside data-driven persona agents, predictive emotional models, a PCG parametric level generator, and a newly defined language for the clear and computable definition of player emotional experiences: ExpREx (Experience Regular Expressions). Using these components, we evolve game levels to match the player experience goals specified using the ExpREx language, aiming to create levels that evoke specific emotional experiences for different subsets of players. The efficacy of our method was validated through a user study involving 101 participants, whose continuous annotations of emotional experience were collected and analyzed to assess the congruence between the actual emotional responses elicited and those targeted by our pipeline. We found that 93.73% of the ExpREx goals targeted were also reported by the user study subjects.

## 1 Introduction

Procedural Content Generation (PCG) is a cornerstone in modern video game development. It allows for the automatic generation of diverse game elements such as levels, characters, objects, and narratives. Traditionally focused on enhancing memory usage, efficiency and variability, PCG has begun to intertwine closely with user experience (UX) and player experience (PX) design.

In this paper, we shift the focus from general PCG to a more targeted approach, where the generation of game content is specifically aimed at eliciting predefined emotional responses from players. Our methodology seeks to construct game levels that consistently provoke intended emotional experiences across all players or tailored to subgroups of players. To be able to define emotional experiences in a way that is both clear and computable, we develop a novel language based on regular expressions named ExpREx (Experience Regular Expressions), which allows the definition of emotional experiences through temporal and spatial dimensions in game settings.

To achieve these targeted emotional responses, our framework incorporates evolutionary algorithms in conjunction with four main components: the aforementioned ExpREx language, which allows us to define emotional experiences; a parametric PCG level generator, which allows us to generate levels based on a set of parameters; persona agents, which are agents that are able to mimic the behaviour of different player personas when interacting with a game and generate play-traces from levels; and predictive emotional models, which allow us to predict the emotional experience a player would have based on a play-trace. A diagram of how these components are used to generate tailored levels can be found on Fig. 1.

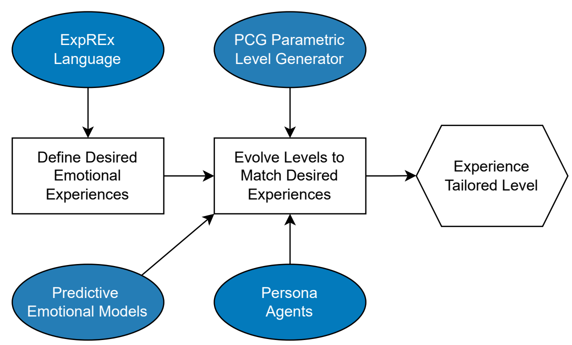



Figure 1: Diagram of the pipeline for generating maps that target specific emotional experiences in players. The 4 main components are represented: the ExpREx language; a parametric PCG level generator; predictive emotional models; and persona agents.

This paper is organized as follows. In Sec. 2 we will give an overview of the several areas of study upon which the work here presented is built upon, along with key papers and books that are especially relevant for the present topic in each of the areas. In Sec. 3, we will present the game used as a test bed for the approach, along with details of the data collection process used to gather the data needed to train the predictive models and the persona agents required for the pipeline. In Sec. 4, we summarize the methodology used to train the predictive models of emotion and the pipeline for the creation of the persona agents. Although these are not the main contributions of this paper, we believe an understanding of the methodologies used here are relevant for understanding the overall approach. Having set out all of the ground work, we present in Sec. 5 one of the main contributions of this paper: the definition of the ExpREx language. In that section, we explain the grammar and regular expression based nature of the ExpREx language, along with showing some examples of how it can be used to define emotional experiences through time and space. In Sec. 6, we present the parametric PCG level generator used and finally explain how the four main components of the model (Fig. 1) are used to evolve experience tailored levels. The results are presented in Sec. 7 paired with a discussion of their implications. In that section we present 6 different levels generated and the efficacy of our approach is validated through an empirical study involving 101 participants. This study tested a generated game level by analyzing player reactions to validate the congruence between intended and actual emotional responses. The generated level matched the desired emotional experience on $93.73\%$ of the tested ExpREx expressions. Finally, in Sec. 8, we conclude this paper.

The main contributions of this work is the definition of a modular and scalable pipeline for the generation of experience-driven game levels along with a novel language, ExpREx, for the clear and computable definition of player experiences.

## 2 Related Work

This work builds upon concepts from several areas. We strive to give a brief overview of the main ones in this section, along with some relevant papers.

### 2.1 Player Modeling and Personas

Player personas, adapted from the original concept of personas by Alan Cooper (1999), are employed in the domain of game design to enhance player modeling. Personas are archetypal user profiles that designers create based on user research to represent distinct, manageable groups of users. In gaming, play personas translate this concept into models that reflect various player types, motivations, and behaviors, thereby aiding game designers in creating more engaging and personalized gaming experiences. By considering different play personas, designers can tailor game mechanics, narratives, and interactions to meet the diverse preferences and needs of their player base.

Togelius et al. (2011) discuss the use of search-based methods to generate game content that adapts to player interaction styles, providing a baseline for understanding different player types. Holmgård et al. (2016) further refines this approach by using evolved heuristics to tailor game experiences to various player archetypes, a method that aligns closely with our use of player personas.

In addition to these foundational works, several other studies have explored the creation and application of player models in game design. Yannakakis and Togelius (2011) have contributed extensively to the field by developing frameworks that adapt game content in real-time based on player experience metrics. Similarly, Drachen et al. (2009) provide insights into clustering techniques for player behavior, which are crucial for developing accurate persona models.

Fernandes et al. (2023b) propose a pipeline for generating persona agents that simulate player behavior by clustering player traces and evolving parametric agents to represent the clusters found. The persona agents used for this paper were created using this pipeline, of which a brief description can be found in Sec. 4.1.

### 2.2 Emotional Modeling in Games

The emotional prediction used in this paper is shaped by the PAD model of emotion (Russell and Mehrabian 1977) and the Core Affect model (Russell 2003).

The PAD model describes emotions along three continuous dimensions: Pleasure (P), Arousal (A), and Dominance (D). This model provides a comprehensive framework for capturing the complexity of human emotions using only three dimensional values, making it highly suitable for predicting player experiences in games.

The Core Affect model, on the other hand, represents basic emotional states as points in a two-dimensional space defined by the dimensions of pleasure-displeasure and activation-deactivation. This model focuses on the underlying structure of emotional experiences which are then interpreted to form more complex emotional states.

The modeling of emotions in game design has been explored through various frameworks and methodologies. The Affect Game AnnotatIoN (AGAIN) dataset introduced by Melhart et al. (2021) and methods discussed by Yannakakis and Togelius (2018) have contributed significantly to the understanding of emotional responses in gaming contexts, with a special focus on the emotional dimension of arousal, utilizing behavioral data to map out the emotional states experienced by players. Barthet et al. (2022) implemented generative models that simulate user interaction within generated game levels to predict player behavior and emotions effectively. Further works and approaches in this area can be found in the book by Karpouzis et al. (2016), which explores how emotion can be measured and defined in the realm of gaming.

Fernandes et al. (2021) propose an agent-based approach for automatic UX testing. Their study highlights the development of agents endowed with basic problem-solving skills and a core affect model. These agents are designed to simulate artificial affective states as they interact with different levels of a game, providing insights into UX without the need for extensive user testing. Building on this foundation, Fernandes et al. (2023a) further present a more advanced modular UX testing agent capable of predicting emotions based on the PAD model of emotion and generating persona-like behavior. This agent mimcs various player types and their emotional responses within a game environment, allowing developers to assess UX design goals and understand user actions and preferences.

### 2.3 Defining Desired Emotional Experiences

One of the primary challenges in experience based automatic level design is the clear definition of emotional experiences in a way that is both computable and verifiable. Traditional methods of playtesting and player feedback are often subjective and difficult to quantify. Addressing this challenge, Ansari et al. (2024) introduce EmoSTL, a Domain-Specific Language (DSL) that extends Linear Temporal Logic (LTL) with spatial and time-interval expressions. EmoSTL captures complex emotional and experiential aspects players undergo, providing a systematic method for articulating and verifying player experience requirements.

Our work builds upon these foundations to define a novel language to define emotional experiences based on regular expressions, ensuring fast computation and seamless integration with most current computing languages, which already support regular expression matching and verification.

### 2.4 PCG and Evolutionary Algorithms in Game Design

Procedural Content Generation (PCG) is a method used to automatically create game content, including levels, items, rules, and even stories, with minimal human input. PCG techniques have been widely adopted in the gaming industry to enhance game replayability, increase content diversity, and reduce development costs. Hendrikx et al. (2013) provide a comprehensive survey of PCG methods used in games, highlighting their applicability across different game elements.

One of the main applications of PCG is in level design, where algorithms generate game maps and environments dynamically. Togelius et al. (2011) discuss search-based PCG, where evolutionary algorithms are utilized to generate levels that can adapt to the player's skill level and preferences, enhancing the player experience. This method aligns with our approach, which integrates PCG techniques with emotional modeling to tailor game levels specifically designed to elicit predetermined emotional responses from players.

The integration of machine learning techniques with PCG has been evolving, particularly in the realm of player-driven level design. Shaker et al. (2012) propose a method that dynamically adapts game difficulty and pacing by leveraging neuroevolution algorithms to assess the player's emotional state in real-time. This method involves constructing computational models from player interaction data, which predict emotional responses such as engagement, frustration, and challenge. The aim is to generate game content in real-time that is tailored to individual player experiences, optimizing player satisfaction and enhancing the immersive quality of the game. This approach exemplifies a shift toward a more nuanced application of evolutionary algorithms in PCG, focusing on emotional impact rather than merely scaling challenges.

The application of evolutionary algorithms for game level design has been extensively studied. Examples include the work by Ecoffet et al. (2021), which employed these algorithms to optimize open-ended exploration in games, and by Garcia-Sánchez et al. (2018), who focused on optimizing gameplay elements in collectible card games. Our research extends these methodologies by integrating evolutionary algorithms with emotional state models to refine game level design to induce specific emotional responses.

Liapis et al. (2015) introduced procedural personas as critics for dungeon generation, representing archetypical player

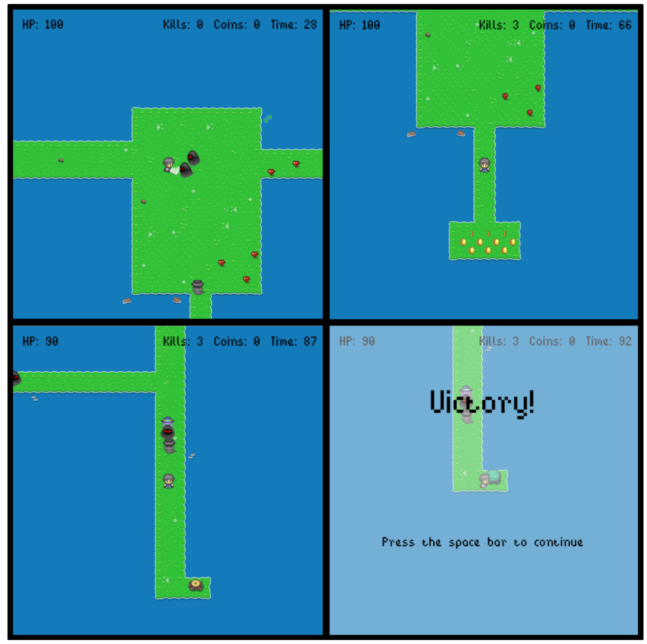



Figure 2: Screenshots from the 2D action-adventure game used in this study. The player navigates a field filled with enemies, health-restoring hearts, and collectible coins (top right). The objective is to reach a tree stump (bottom left), which triggers the tree growth and ends the level upon contact (bottom right). Players use a sword to combat enemies (top left). If the player's health depletes entirely from enemy contact, the game ends in failure.

behaviors to evaluate and optimize dungeon layouts. Fernandes et al. (2021) further demonstrated how procedural content generation (PCG) can adapt to player personas through evolutionary algorithms. Their architecture uses persona agents and simple experience metrics to evolve game levels tailored for specific player personas. This approach, tested in their game "Grave Rave," successfully adapts to various personas, ensuring the generated levels are not just general optimizations but are specifically tailored for each persona.

In conclusion, our work synthesizes and extends existing research in player modeling, experience definition, emotional prediction, procedural content generation, and evolutionary game design to establish a new approach for creating experience tailored game levels. This contributes both to academic research and practical applications in game design, aiming to enhance user engagement through tailored emotional experiences.

## 3 Game and Dataset

To validate our approach, we utilized a 2D action-adventure game (Fig. 2), drawing inspiration from classic top-down games such as Legend of Zelda. In this game, the player controls a character equipped with a sword, capable of collecting coins (representing wealth) and hearts (representing health), with ghosts as opponents to fight. The objective of each level is to locate the stump of a tree; upon reaching

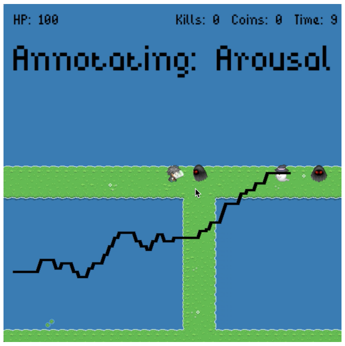



Figure 3: Annotation process used to collect emotional data from participants. After playing a level, the players would see a repetition of their playthrough and control a black line in the middle of the screen, moving it upwards or downwards to report an increase of decrease in the assigned emotional dimension, respectively.

the stump, the tree grows, signifying the level's completion. This game was selected due to its simplicity, while at the same time enabling a wider range of player behaviors and experiences than the simpler games often used in PCG research.

We engaged 91 participants, primarily first-year Psychology students, who played through three hand-designed levels of the game. Participants were instructed on the game mechanics and allowed to play freely without time restrictions. During their playthroughs, we recorded all their actions in the game. After completing each level, players were asked to self-annotate their emotional experience based on two emotional dimensions: *Pleasure* and *Arousal*. Further explanation of these dimensions can be found in Sec. 2.2 and Sec. 4.2.

Each participant was randomly assigned one of the dimensions to annotate and provided with a clear explanation of the assigned dimension. They proceeded with the annotation only after confirming their understanding. The annotation method was conducted post-gameplay and was continuous, akin to the method described in Lopes et al. (2017) and Fernandes et al. (2023a). Participants watched a video of their gameplay and used the keyboard's up and down arrows to rate their emotions, indicating increases or decreases in the assigned dimension. A line chart displayed the evolution of their emotional state over the video. A snapshot of the annotation process can be found in Fig. 3.

The dataset, comprising detailed playthrough actions and emotional annotations, was used to train the persona agents and the predictive models described in this paper.

# 4 Predicting Emotional Experiences from Levels

To predict the emotional experience that players will have when interacting with a previously unseen level, we need to solve two problems: 1) - we need to have agents that mimic the behaviour real players would have when interacting with the level; 2) - we need a method to predict the emotional experience of a player based on a level and the corresponding gameplay trace. In this section, we briefly cover the approach used to solve both these problems.

## 4.1 The Persona Agent Pipeline

Persona agents are grounded in the concept of 'play-personas' from Human-Computer Interaction, and are designed to mimic the behavior of different player types identified through data-driven methods.

The generation of persona agents is a systematic process involving the collection, analysis, and modeling of player data to create agents that behave like real players. The process is detailed as follows:

1. **Data Collection:** The initial step involves gathering extensive play-traces, which are sequences of actions and decisions made by players during gameplay, from different players. They serve as the foundational data for generating persona agents. The data collection methodology for the creation of the persona agents used in this paper is described in Sec. 3.
2. **Persona Creation from Play-traces:** Using clustering algorithms, play-traces are analyzed and grouped based on similarities in gameplay behavior. Each cluster corresponds to a distinct player persona, reflecting a unique set of behaviors and preferences that characterize a segment of the game's audience. To define these clusters, a distance metric that defines the behavioural similarity between different play-traces is defined. For this work, the Levenshtein distance (Levenshtein et al. 1966) between string representations of the high granularity sequence of actions taken by players was used.
3. **Parameter Optimization:** Persona agents are then evolved through an evolutionary algorithm where the parameters governing agent behaviors are iteratively adjusted. The aim is to optimize these parameters so that the actions of the agents closely match the clustered play-traces. The fitness of each agent is thus the level of similarity between the trace produced and the traces of the cluster that the agent is meant to represent.

A diagram showing the pipeline for persona agent generation can be found in Fig. 4. The persona agents used for this paper followed the same principles and algorithms as the ones described in the work of Fernandes et al. (2023b), where a more detailed explanation of the persona agent pipeline can also be found.

## 4.2 Predicting Player Experience

We have adopted a machine learning approach to develop a predictive model of player emotions based on the three-dimensional PAD model of emotion and the two dimensional model of core affect. Our model thus predicted two emotional dimensions, Pleasure and Arousal. These dimensions are present in both the PAD model and the core affect model, being able to jointly represent a plethora of basic emotional states. We have decided not to use the full PAD model as the Dominance emotional dimension of the model has been shown to be very hard to accurately predict (Fernandes, Lopes, and Prada 2023a).

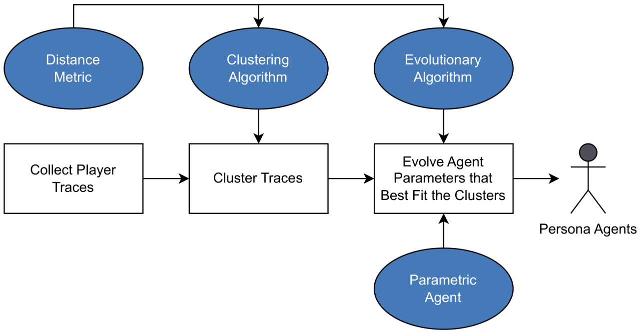



Figure 4: Diagram of the persona agent pipeline, identifying the 4 main components that need to be implemented: a distance metric; a clustering algorithm; an evolutionary algorithm; and a parametric agent.

To train our predictive model, we used the data collected as described in Sec. 3 and used the random forests algorithm (Breiman 2001) to model the relationship between the collected game data and the changes in emotional states, aiming to accurately predict future emotional changes from similar events.

A different predictive model had to be trained for the Pleasure and Arousal emotional dimensions. As such, we achieved a different accuracy for each dimension. The accuracy was based on the correct classification of 1 second slices using the "leave one out" method, where we measured the average classification accuracy of the slices from one of the players with a model trained with the data from all the other players. For the Pleasure dimension, we were able to achieve an accuracy of 72.8%. For the Arousal dimension, we were able to obtain a slightly better 73.1% of accuracy.

The trained model enabled us to predict emotional responses of players during new gameplay sessions. This predictive capability allowed us to predict the expected emotional response of players when given a play trace. Using the previously described Persona Agents, we could thus generate traces and then predict the expected emotional response of players given a level.

## 5 ExpREx: Defining Player Experiences

In this paper, we introduce ExpREx (Experience Regular Expressions), a novel language for encoding and analyzing player experiences in video games. Drawing inspiration from regular expressions, commonly used in pattern matching with strings, ExpREx is tailored to represent complex emotional changes that players undergo during gameplay.

### 5.1 Fundamental Components of Regular Expressions

Regular expressions serve as a foundational tool for pattern matching and string manipulation across various computing environments. Grasping the key elements of regular expressions is crucial for understanding the ExpREx language.

Regex patterns are constructed using a mix of ordinary characters and special metacharacters. The latter are essential for denoting operations or specifying constraints in patterns:

- **Metacharacters:** Special symbols in regex that enable advanced matching capabilities include:
  - **.** (dot) – Matches any single character except a newline.
  - **ˆ** – Anchors the regex to the start of a string.
  - **$** – Anchors the regex to the end of a string.
  - ***** – Matches zero or more repetitions of the preceding element.
  - **+** – Matches one or more repetitions of the preceding element.
  - **?** – Makes the preceding element optional.
  - **{}** – Specifies a range of repetitions for the preceding element.
  - **[]** – Defines a set of characters to include in a match.
  - **|** – Represents a logical OR, allowing alternative patterns to be matched.
- **Character Classes:**
  - **[a-z]** – Matches any lowercase letter from a to z.
  - **[A-Z]** – Matches any uppercase letter from A to Z.
  - **[0-9]** or **\d** – Matches any digit.
  - **\s** – Matches any whitespace character.
  - **\S** – Matches any non-whitespace character.
  - **\w** – Matches any word character, useful for text-based pattern matching.
  - **\W** – Matches any non-word character.
- **Escape Character (\):** Used to indicate that the following character should be treated as a literal.

### 5.2 ExpREx Format and Implementation

The ExpREx language is a novel approach designed to encode emotional experiences within video game environments. Drawing on the foundational principles of regular expressions commonly used in text processing, ExpREx allows for the detailed specification of emotional trajectories that players should experience while interacting with a game. The language leverages the syntax and mechanics of regular expressions to create a flexible and powerful tool for mapping out complex emotional patterns over time and space within game levels.

Central to the design of the ExpREx language is the concept of using a structured format to detail the emotional states and changes a player undergoes during gameplay. This format is akin to the way regular expressions operate by matching patterns within strings, but instead, it matches patterns within the gameplay experience.

The ExpREx language encodes an experience as a sequence of words in a specific format, each one representing one second and having information about the location and the emotional trajectory of the player. For example, a gameplay of 60 seconds would be encoded by an ExpREx sentence of 60 words, the first one containing information about the first second of the experience, and the last one about the last second of the experience. Each word has the following format:

```
R[0-9]*A[idsn]P[idsn]
```

where:

- **R** denotes the room or in-level location that the player is in.
- **A** represents Arousal changes, with the following modifiers:
  - **i**: Increase
  - **d**: Decrease
  - **s**: Steady
  - **n**: No data
- **P** stands for Pleasure changes, with the same modifiers as Arousal.

By sequencing these words, we are able to transform an emotional experience into a string which contains temporal, spacial and emotional information. Once this information is in string format, we can then leverage the flexibility and precision of regular expressions to encode desired emotional journeys during gameplay. By utilizing regex constructs like quantifiers, character classes, and logical operators, ExpREx can effectively describe complex sequences of gameplay experiences.

### 5.3 Examples of ExpREx Expressions

To illustrate the application of the ExpREx language, consider the following scenarios and their corresponding expressions:

- **Example 1**
  - **Natural Language**: Arousal should steadily increase throughout all rooms of the level until the player reaches room 5, where arousal should decrease and pleasure increase.
  - **Regular Expression**: $(R[0-9]*AiP.)+(R5AdPi)+$
- **Example 2**
  - **Natural Language**: In Room 1, Arousal must increase and Pleasure decrease. Then either Arousal increases in Room 2 or 3 or Arousal decreases in Room 4. Finally, in Room 5, Pleasure should remain steady.
  - **Regular Expression**: $(R1AiPd)+((R[23]AiP.)|(R4AdP.))+(R5A.Ps)+$
- **Example 3**
  - **Natural Language**: The player's arousal must increase continuously for at least ten seconds, followed by an increase in pleasure for a minimum of another ten seconds, and finally a decrease in arousal for a minimum of ten seconds.
  - **Regular Expression**: $.*(R[0-9]*AiP.)\{10\}.*(R[0-9]*A.Pi)\{10\}.*(R[0-9]*AdP.)\{10\}.*$

Each of these examples demonstrates how ExpREx can be used to succinctly encode the expected emotional trajectories in a way that is both computable and flexible.

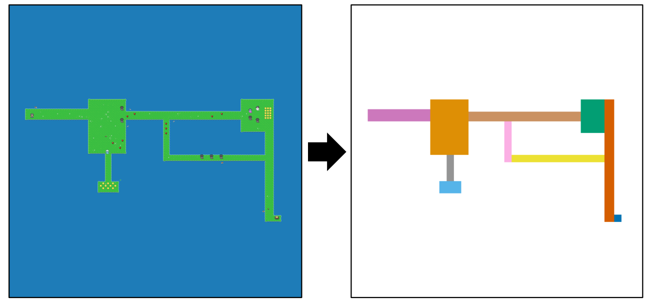


Figure 5: Example of the automatic division of a map into different rooms and corridors based on the topology. Each differently coloured region is assigned a different number which can then be referenced using the ExpREx language

### 5.4 ExpREx Validation

To validate whether a play-trace matches the encoded experience in an ExpREx expression, we first need to transform the play-trace into a sequence of $R[0-9]*A[idsn]P[idsn]$ words, one for each second of the play-trace. Using our persona agents and our emotional predictive models, we have all the information regarding the location and the emotional trajectory at each given second of a trace. We thus need only transform this information into the correct format. The location information gathered during the interaction of the agent with the level were coordinates. The levels were generated and had no pre-defined locations. We thus designed an algorithm that analysed a generated map and divided it into several locations based on the topology of the map, numbering each individual corridor and room found. An example of this division of a map can be found in Fig. 5. Having this division, we now had a general location for each second of the game which was of a much lower granularity than the in game coordinates and thus more meaningful for the experiences we wished to encode. As for the encoding of the emotional data, we had continuous annotations for both the Arousal and Pleasure emotional dimensions. As such, we needed only to access at each second whether that dimension increased or decreased and save that data on the specified format. By thus encoding the information regarding the location and the emotional trajectory, we are able to create sentences that can then be matched to ExpREx expressions.

## 6 Evolving Levels for Experience

This section delineates the methodology employed to evolve game levels that adhere to predefined emotional experiences as specified by our ExpREx language. The approach integrates evolutionary algorithms to evolve levels that elicit specified emotional responses across diverse player profiles.

### 6.1 Genomic Parameters in Level Generation

The genome for the evolutionary algorithm in our level generation process consists of several parameters which directly influence the structure and complexity of the generated game maps. These parameters, acting as genes within our evolutionary framework, are essential for tailoring the level design to meet the desired emotional experiences encoded in ExpREx.

The level generation algorithm operates through a pointer that traverses the predefined map space, systematically creating corridors as it progresses. This pointer, guided by several parameters, determines the path's trajectory, frequency of turns (rotation frequency), and the intervals at which rooms are spawned (room frequency). As the pointer moves, it periodically decides whether to change direction based on the rotation frequency or to introduce a room depending on the room frequency and the maximum number of rooms allowed. Corridor widths and lengths vary within specified ranges, allowing for diverse path configurations. The decision-making process for these changes is influenced by the set parameters, which include corridor and room sizes, the minimum and maximum lengths for corridors, and the likelihood of adding gameplay elements such as enemies, coins, and health items. The algorithm ensures that each path and room placement respects the boundaries of the canvas size and adheres to the constraints of total length. This procedural method allows for the dynamic creation of game maps that are not only varied and interesting but also tailored to enhance player experience through structured yet unpredictable level design.

The parameters and their roles in level creation are detailed below:

- **Canvas Size:** Defines the dimensions of the game map. Larger canvas sizes allow for more complex levels with diverse gameplay elements.
- **Total Length:** Specifies the total length of paths (corridors) generated, which determines the expansiveness of the level.
- **Rotation Frequency:** Influences the frequency at which the direction of path changes, impacting the level's layout complexity.
- **Room Frequency:** Determines how frequently rooms are placed along the path, contributing to the structural diversity within the level.
- **Maximum Number of Rooms:** Caps the total number of rooms that can be generated in a level, controlling the balance between open spaces and enclosed areas.
- **Corridor Width (Minimum and Maximum):** These parameters set the range for the width of the corridors, affecting how spacious or cramped the paths feel.
- **Room Size (Minimum and Maximum):** Dictate the size range for rooms, influencing their utility and the strategic options they offer.
- **Corridor Length (Minimum and Maximum):** These define the length range for single stretches of corridor between turns or rooms, affecting the pacing of the level.
- **Number of Enemies, Coins, and Healths:** These parameters specify the quantities of enemies, coins, and health pickups placed within the level, directly impacting the level's difficulty and reward structure.

Each of these parameters can be mutated or crossed over in the evolutionary process to produce new levels with varied characteristics.

## 6.2 Fitness Function

Persona agents, as described in Sec. 4.1, are employed to interact with the initial level designs. We used five different persona agents, representing five different persona clusters found in our dataset. The fitness function within our evolutionary algorithm is designed to evaluate how well a level meets the emotional trajectory requirements specified by a set of ExpREx expressions across all personas.

The fitness score for a level is determined by assessing each persona's emotional response against the entire set of ExpREx expressions. A match occurs when a persona's emotional trajectory aligns with any of the ExpREx-defined trajectories. The fitness score is thus a sum of the matches across all personas and all ExpREx expressions:

$$\text{Fit(Level)} = \sum_{i=1}^{N} \sum_{j=1}^{M} \text{Match}(Persona_i, \text{Level}, \text{ExpREx}_j) \tag{1}$$

where:

- Match is a function that returns 1 if the persona's emotional response matches the $j$-th ExpREx for the level, and 0 otherwise.
- $N$ is the total number of personas tested.
- $M$ is the total number of ExpREx expressions.

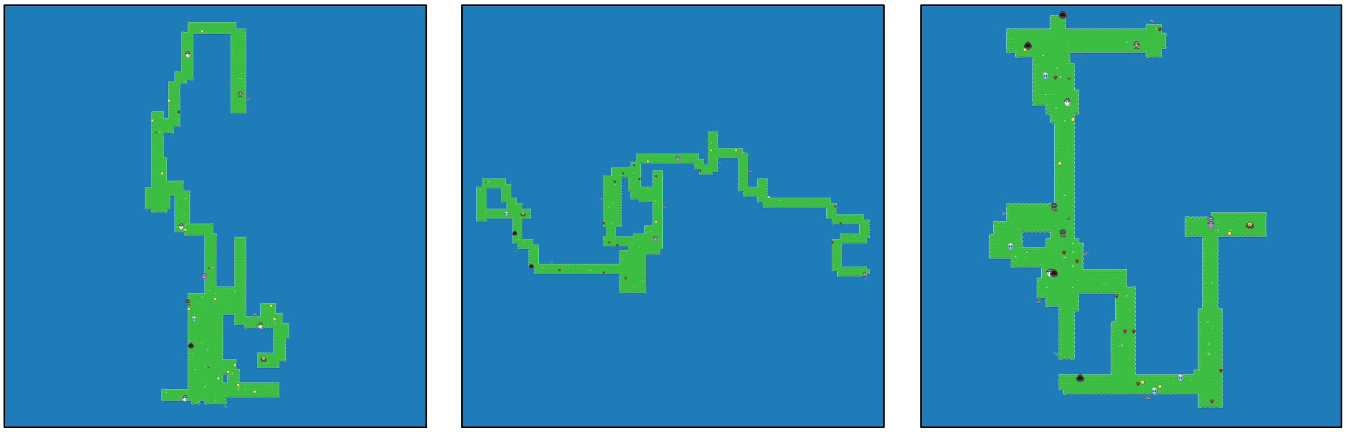
This fitness evaluation ensures that levels which elicit the desired emotional responses across a broader set of ExpREx expressions and personas are deemed more suitable. Levels achieving higher fitness scores are more likely to survive and propagate through the evolutionary process, leading to game environments that are emotionally engaging for a diverse range of players and scenarios.

By employing a fitness function that values the breadth of emotional alignment across both player personas and ExpREx expressions, the developed levels are optimized for comprehensive emotional impact, ensuring that the levels resonate well with different player personas.

## 6.3 Evolutionary Algorithm

The evolutionary algorithm used consisted of the following steps:

1. **Setup:** A population of 100 maps is generated with random parameters, following the description of Sec. 6.1.
2. **Simulation and Fitness Evaluation:** Each one of the levels is played by the five chosen persona agents and the level's fitness is calculated according to Sec. 6.2
3. **Selection:** The top $20\%$ of the population with the highest fitness remains in the population and will serve as the parents for the next generation of levels. The bottom $80\%$ of the population with the lowest fitness is discarded.
4. **Populate:** Pairs of two levels that are still in the population are chosen at random to create a new level by randomly mixing the parameters of both parent levels with equal probability. With a probability of $10\%$, each parameter can suffer a mutation and assume a random value in the pre-defined range of possible values.



a) b) c)

Figure 6: Three levels evolved to maximize different sets of ExpREx expressions for all 5 personas, ensuring a similar experience for different play types. The expressions maximized for each level are the following: **a):** "$.*(R[0-9]*AiP.)\{3\}.*$", "$.*(R[0-9]*AdP.)\{10\}.*(R[0-9]*AiP.)\{3\}.*(R[0-9]*AdP.)\{10\}.*$", "$.*(R[0-9]*AiP.)\{10\}.*$" ; **b):** "$.*(R[0-9]*AiP.)\{3\}.*(R[0-9]*A.Pi)\{3\}.*(R[0-9]*AdP.)\{3\}.*$", "$.*(R[0-9]*A.Pd)\{3\}(R[0-9]*A.Pi)\{3\}.*$", "$.*(R[0-9]*AdPd)\{5\}.*$" ; **c):** "$.*(R[0-9]*AiP.)\{15\}.*$", "$.*(R[0-9]*AiP.)\{5\}.*$", "$.*(R[0-9]*AiP.)\{10\}.*$".

5. **Repetition:** Steps 2 to 5 were repeated for 30 generations. After this, the level with the highest fitness in the population was chosen as the generated level.

## 7 Results and Discussion

We begin this section by showcasing three different levels that were evolved to match sets of ExpREx expressions for five different personas. These levels can be found in Fig. 6, along with the used ExpREx on the figures' descriptions. These levels differ not only on topology but also on the density and location of the different in game objects and enemies.

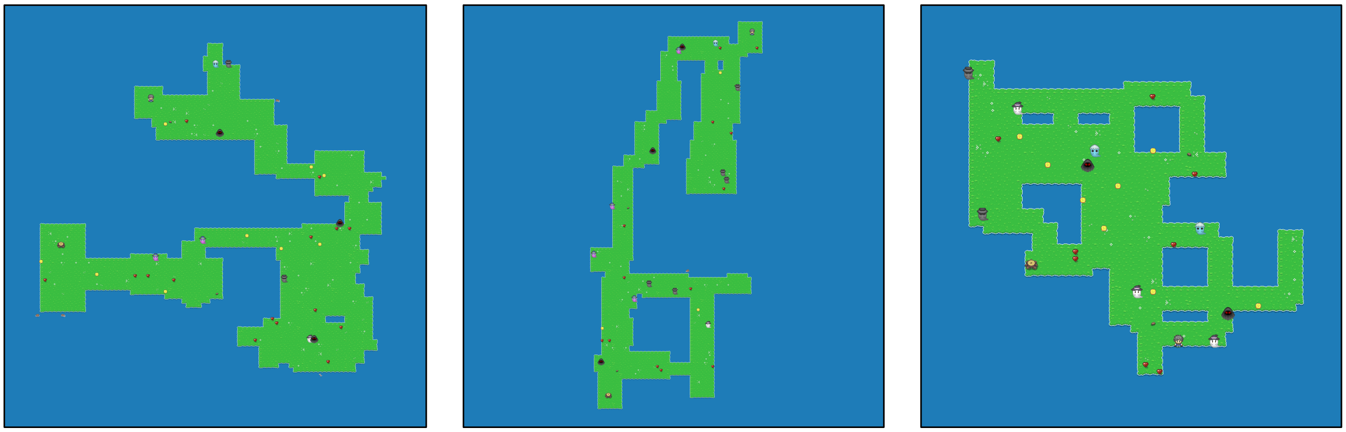
To validate the effectiveness of our approach, we conducted a user study involving 101 participants. Players used after the fact continuous annotation as described in Sec 3 and shown in Fig.3 to annotate their emotional traversal through the level shown in Fig. 6 b). We then matched the obtained emotional annotations with the defined ExpREx expressions to assess whether players were indeed experiencing the predefined emotional traversals encoded.

The level had been evolved to maximize the 3 different ExpREx expressions descrbed in the description of Fig. 6 b). For this validation, the perfect end goal would be for each of the 101 players to experience all 3 ExpREx expressions. This would mean that 303 expressions (3 for each player) should be successfully matched with the reported experience of their corresponding players. In this user study, we were able to successfully match 284 of the possible 303 expressions, which represents an accuracy of $93.73\%$. This is a promising result for the approach, especially given the known accuracy of both the emotional predictive models to be $\approx 73\%$ (Sec. 4.2). The use of more accurate emotion prediction models along with more diverse persona agents would be expected to further improve the emotional accuracy of the generated levels. This user study, although promising, is only an initial validation of the approach. We believe the pipeline should be further validated in more levels and in different games.

These generated levels were evolved to elicit the same ExpREx defined emotional patterns in different player personas. It is not surprising then that these levels, although allowing for some degree of choice, guide the players to take similar paths. To design levels that elicit different emotional experiences for different types of players, one can modify the fitness function to match ExpREx expressions that are unique for each type of player. We did so, using five different ExpREx expressions and assigning each one to a player persona. We generated three levels, changing the expression assigned to each persona between them. The resulting levels can be found in Fig. 7. These generated levels no longer guide the players to follow such a strict path as the levels found in Fig. 6, allowing for a higher degree of behavioural choice and thus allowing each play persona to have a different experience while playing the level.

The capabilities and accuracy of this approach merit further testing and validation. The results here presented show promise, but they are insufficient beyond a proof of concept and further user studies in several different levels and even several different games are required for a robust evaluation of the pipeline.

The evolutionary algorithms used for level design could be expanded to include multi-objective optimization, including a broader range of factors beyond emotional impact, such as player engagement and difficulty. This would enable a more holistic approach to game design, where emotional responses are considered alongside other critical aspects of



a) b) c)

Figure 7: Three levels evolved to maximize a single and unique ExpREx expression for each of the 5 personas, evolving a level tailored for a different experience for each different play person. The same five expressions were used for the three levels, the difference being that they were assigned to different personas in each level. The expressions used were the following: "$.*(R[0-9]*AdPd)\{5\}.*$"; "$.*(R[0-9]*A.Pi)\{5\}.*$"; "$.*(R[0-9]*A.Pd)\{3\}(R[0-9]*A.Pi)\{3\}.*$"; "$.*(R[0-9]*AiP.)\{3\}.*(R[0-9]*A.Pi)\{3\}.*(R[0-9]*AdP.)\{3\}.*$"; "$.*(R1*AiP.)\{3\}.*$".

player experience.

The dependence on the ExpREx language to define and predict emotional trajectories introduces limitations related to the interpretability and flexibility of these definitions. While ExpREx provides a computable and structured way to encode emotional responses, it might not encompass all possible emotional states that arise during gameplay, particularly those that do not fit neatly into predefined categories. Future versions of ExpREx could benefit from integrating more adaptive and context-aware elements that can dynamically adjust to the unique and evolving gameplay situations, thus offering a more granular and accurate mapping of player emotions. They could also include other experience metrics besides emotions, such as perceived difficulty.

For this work, we took only into consideration play personas in regards to behaviour. Future work could also explore the role of experience personas, that is, players that experience differently the same set of circumstances. For one player, being surrounded by enemies might be a positive experience because it wishes to fight them, whilst for another, that might be a negative experience because it wishes to avoid fighting.

Finally, the ethical implications of manipulating player emotions through game design merit careful consideration. As technology advances, the ability to influence player psychology so directly raises questions about consent, privacy, and the potential for emotional manipulation. Establishing ethical guidelines and best practices for emotion-driven game design will be essential as these technologies evolve.

## 8 Conclusion

This paper presents a pipeline for evolving game levels aimed at evoking predefined emotional experiences in players. Our approach integrates four main components: the ExpREx language, a parametric PCG level generator, persona agents, and predictive emotional models. Each component plays a crucial role in achieving the overarching goal of creating emotionally tailored game levels.

The ExpREx language is a novel contribution that leverages regular expressions to define and analyze emotional experiences in a clear and computable manner. This language allows for the precise encoding of complex emotional trajectories over time and space within game levels, providing a robust tool for experience-driven game design.

An initial validation of the approach was conducted through an empirical study involving 101 participants. The results showed that the generated level was able to evoke the intended emotional experiences with a high degree of accuracy, achieving a match rate of 93.73%. This finding supports the effectiveness of the proposed methodology in creating game levels that align closely with the predefined emotional trajectories

Overall, the methodology outlined in this paper provides a structured approach to experience-driven game design, enabling the creation of game levels that are tailored to evoke specific emotional experiences in players. This work lays the groundwork for further exploration in the intersection of PCG, player modeling, and emotional prediction, offering insights and tools for both academic research and practical applications in the gaming industry.

## Acknowledgments

This work was supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with references UIDB/50021/2020 and 2020.05865.BD.

## References


Ansari, S. G.; Prasetya, I. S. W. B.; Dastani, M.; Dignum, F.; and Keller, G. 2024. EmoSTL: Formal Spatial-Temporal Verification of Emotion Specifications in Computer Games. *Proceedings of the 17th IEEE International Conference on Software Testing, Verification and Validation (ICST)*.

Barthet, M.; Khalifa, A.; Liapis, A.; and Yannakakis, G. 2022. Generative personas that behave and experience like humans. In *Proceedings of the 17th International Conference on the Foundations of Digital Games*, 1–10.

Breiman, L. 2001. Random forests. *Machine learning*, 45: 5–32.

Cooper, A. 1999. The inmates are running the asylum. In *Software-Ergonomie'99*, 17–17. Springer.

Drachen, A.; Canossa, A.; and Yannakakis, G. N. 2009. Player modeling using self-organization in Tomb Raider: Underworld. In *2009 IEEE symposium on computational intelligence and games*, 1–8. IEEE.

Ecoffet, A.; Huizinga, J.; Lehman, J.; Stanley, K. O.; and Clune, J. 2021. First return, then explore. *Nature*, 590(7847): 580–586.

Fernandes, P. M.; Jørgensen, J.; and Poldervaart, N. N. 2021. Adapting procedural content generation to player personas through evolution. In *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, 01–09. IEEE.

Fernandes, P. M.; Lopes, M.; and Prada, R. 2021. Agents for automated user experience testing. In *2021 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, 247–253. IEEE.

Fernandes, P. M.; Lopes, M.; and Prada, R. 2023a. Data Driven Agents for User Experience Testing. In *International Conference on Videogame Sciences and Arts*, 3–18. Springer.

Fernandes, P. M.; Lopes, M.; and Prada, R. 2023b. Persona agents from playtraces and emotion. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 19, 215–223.

García-Sánchez, P.; Tonda, A.; Mora, A. M.; Squillero, G.; and Merelo, J. J. 2018. Automated playtesting in collectible card games using evolutionary algorithms: A case study in hearthstone. *Knowledge-Based Systems*, 153: 133–146.

Hendrikx, M.; Meijer, S.; Van Der Velden, J.; and Iosup, A. 2013. Procedural content generation for games: A survey. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 9(1): 1–22.

Holmgård, C.; Liapis, A.; Togelius, J.; and Yannakakis, G. N. 2016. Evolving models of player decision making: Personas versus clones. *Entertainment Computing*, 16: 95–104.

Karpouzis, K.; and Yannakakis, G. N. 2016. *Emotion in Games*. Springer.

Levenshtein, V. I.; et al. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, 707–710. Soviet Union.

Liapis, A.; Holmgård, C.; Yannakakis, G. N.; and Togelius, J. 2015. Procedural personas as critics for dungeon generation. In *Applications of Evolutionary Computation: 18th European Conference, EvoApplications 2015, Copenhagen, Denmark, April 8-10, 2015, Proceedings 18*, 331–343. Springer.

Lopes, P.; Yannakakis, G. N.; and Liapis, A. 2017. Ranktrace: Relative and unbounded affect annotation. In *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*, 158–163. IEEE.

Melhart, D.; Liapis, A.; and Yannakakis, G. N. 2021. The Affect Game AnnotatIoN (AGAIN) Dataset. *arXiv preprint arXiv:2104.02643*.

Russell, J. A. 2003. Core affect and the psychological construction of emotion. *Psychological review*, 110(1): 145.

Russell, J. A.; and Mehrabian, A. 1977. Evidence for a three-factor theory of emotions. *Journal of research in Personality*, 11(3): 273–294.

Shaker, N.; Yannakakis, G. N.; and Togelius, J. 2012. Towards player-driven procedural content generation. In *Proceedings of the 9th conference on Computing Frontiers*, 237–240.

Togelius, J.; Yannakakis, G. N.; Stanley, K. O.; and Browne, C. 2011. Search-based procedural content generation: A taxonomy and survey. *IEEE Transactions on Computational Intelligence and AI in Games*, 3(3): 172–186.

Yannakakis, G. N.; and Togelius, J. 2011. Experience-driven procedural content generation. *IEEE Transactions on Affective Computing*, 2(3): 147–161.

Yannakakis, G. N.; and Togelius, J. 2018. Artificial Intelligence and Games. In *Springer*.