

Procedurally generating rules to adapt difficulty for narrative puzzle games

1st Thomas Volden
Center for Digital Play
IT University of Copenhagen
Copenhagen, Denmark
thvo@itu.dk

2nd Djordje Grbic
Center for Digital Play
IT University of Copenhagen
Copenhagen, Denmark
djgr@itu.dk

3rd Paolo Burelli
Center for Digital Play
IT University of Copenhagen
Copenhagen, Denmark
pabu@itu.dk

Abstract—This short paper focuses on procedurally generating rules and communicating them to players to adjust the difficulty. This is part of a larger project to collect and adapt games in educational games for young children using a digital puzzle game designed for kindergartens. A genetic algorithm is used together with a difficulty measure to find a target number of solution sets and a large language model is used to communicate the rules in a narrative context. During testing the approach was able to find rules that approximate any given target difficulty within two dozen generations on average. The approach was combined with a large language model to create a narrative puzzle game where players have to host a dinner for animals that can't get along. Future experiments will try to improve evaluation, specialize the language model on children's literature, and collect multi-modal data from players to guide adaptation.

Index Terms—procedural content generation, large language models, genetic algorithm, learning environment, early childhood education

I. INTRODUCTION

Research on serious games has shown that when done correctly an educational game can increase a child's motivation to learn and by extension optimize the learning outcome [6].

The concept of adaptive learning can be dated back to as early as 1970 [2]. The core principle is to track the performance of the learner and to adapt the learning content to improve this performance.

Due to their ability to customize content for the player, digital games have the potential to maximize learning. This potential has been explored in different fields, such as medical training and schools with mixed results [15] [13]. Although the use of serious games to improve cognitive capacities has yielded promising results [10], the potential of adapting to the player may not have been fully realized [12].

One promising direction in the field, multi-modal learning analytics, focuses on fusing multiple sources of data to build more comprehensive learning profiles [7]. This approach has shown the potential to improve the modelling of the learner especially for children. There are however limited studies in the area, due to technical challenges with multi-model data collection for children [3].

Several studies have investigated the interplay between multi-modal player profiling and procedural content generation and adaptation in games [14].

A few recent have explored the application of procedural content generation methods for learning games, but the studies have been confined within on-screen fully digital experiences [5] or on-screen simulation of physical experiences [11].

This project approaches procedural content generation for physical tile-based games in collaboration with YOLI¹ which specializes in tangible digital learning games for children in kindergartens. Specifically, this project case studies the YOLI Board, a physical game device developed by YOLI, which consists of a board and a set of tiles with pictures. The board contains five spaces for the player to place tiles which the board then registers and interacts with by shaking or throwing off a tile. Games are packaged as a set of thirty tiles and come with various rules and themes to train and support the development of young children.

The digital nature of the YOLI Board allows it to collect anonymous player behaviour information and has the potential to adapt to such information. This makes it a promising candidate as a test bed for dynamic difficulty adjustment. Even with the constraints of a fixed set of tiles, it is possible to adapt the difficulty by complicating or simplifying rules.

The goal of this project is to answer this research question. How can player behaviour be gathered and used to procedurally generate content and adapt the learning experience to improve the social and cognitive skills of children? This short paper will focus on the procedural content generation part of the research question and try to answer a more specific question in order to contribute to the final answer. How can a game with a fixed set of tiles designed for kindergarteners dynamically adapt difficulty by changing and communicating rules in such a way that the children are challenged but not overwhelmed? It is essential to recognize that the rules themselves, as well as how they are communicated, add to the difficulty level of a game.

Recent studies have shown promising results by utilizing large language models to produce contextual rationalizations with the use of special prompt techniques [9] [4]. This project will use large language models to experiment with different ways to communicate rules and how to use narrative context to rationalize rules that may otherwise be hard to comprehend.

¹<https://www.playyoli.com/>

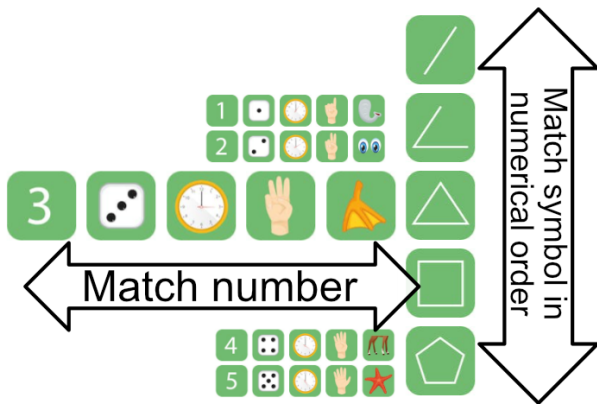


Fig. 1. Example of a tile game with thirty tiles, where a player has to match tiles that represent the same number or the same symbol in numerical order.

If all rules are produced at random to find a set of rules with a slightly higher difficulty level, the player would have to disregard previous rules and learn a set of rules with that complexity level. To avoid this, rules should only be altered slightly, for example by adding a new rule, eliminating a rule, or modifying the parameters of a rule.

II. YOLI BOARD AND SIMULATOR

The YOLI company has specialized in child-friendly educational games and has delivered numerous YOLI Boards to kindergartens in the north of Europe. Under a child-friendly posterior, the YOLI Board is equipped with wireless chip readers which are used to read and match digitally imprinted information on the tiles. The board is able to shake or entirely throw off individual tiles using magnets, which produces tangible feedback for the players.

The goal of most games is to place five tiles on the board that complies with the rules, which are mainly to match certain attributes of the tiles. An example of this is the "Fun with numbers" game, shown in figure 1, which features thirty tiles representing a number. The rule of the game is to match tiles that represent the same number or match tiles of the same symbol in numerical order.

The board can connect with a mobile phone where a dedicated app can assist with audio and other more advanced game features. The mobile app is also used for data collection and transfer from the YOLI Board.

A software simulator of the physical board had to be developed to test and evaluate games and rules efficiently. The simulator was developed in Python and can be operated manually or via the gym environment for training and testing agents [1].

The simulator was built around the architectural decision to separate the board and games, which means that the board simulation can be initialized with games that inherit from a tile game-specific abstract class. The board implementation maintains state information above what tiles are placed where and handles user input and game evaluation output. Tile games provide a list of tiles and a method to evaluate a solution set

of tiles and return a list to accept, reject or ignore individual tiles.

III. PROCEDURAL RULE GENERATOR

The simulator provides tools to assist in the creation of custom tile-based games and accommodate game adaptations with procedural rule generators and rule mutations for genetic algorithms.

The procedural rule generator generates rules based on rule concepts and parameters. A rule concept should limit the number of solutions with precision and be formulated in a way that can reflect a conflict in a narrative context. To accomplish this the following rule concepts were defined, with some designed to exclude a large number of tiles while others exclude a small but specific set of tiles:

- Exclude any where *property* is equal to *value*.
- Exclusively for any where *property* is equal to *value*.
- Only the ones where the value of *property* match.
- Those with *property* set to *value* cannot be adjacent to those with *property2* set to *value2*.
- There can only be *number* with *property* set to *value*.

The procedural rule generator picks a rule concept from the list at random and samples the parameters (*property* and *value*) from a list of tiles, like the latest set of tiles that solved the puzzle. For example, a game featuring vehicle tiles could sample the *property type* and a *value* among *car*, *bus*, or *motorcycle*. Combined with the rule concept this can form the rule "Vehicles of type bus can not be next to vehicles with type motorcycle".

Multiple rules can be combined to add more precision. This is done using a special rule concept called a composite rule, to which child rules can be added or removed. The composite rule evaluates a solution on each of its child rules and merges the result using pair-wise comparison. A tile is rejected if at least one of the child rules rejects the tile.

The difficulty is determined by the number of allowed solutions, consisting of sets of five tiles that are not rejected by the rules. With a maximum of 142,506 solutions for a 30-tile game where the order is irrelevant, this measure indicates an increasing challenge level, from trivial to impossible, as the number of solutions decreases.

The evaluation of difficulty and procedural rule generator have been combined in a genetic algorithm to search for a target difficulty. Rule concepts are mutated by changing one or more random parameters for the rule concept. The composite rule concept is also able to add and remove a random rule as part of its mutation.

IV. FRAMING RULES IN A NARRATIVE CONTEXT

Procedurally generated rules, such as the vehicle example given earlier, naturally result in the question "why?". To minimize this, a large language model is used to communicate rules, which at the same time allows for more creative delivery than the rule generator can provide. The current approach assembles a prompt that contains information about the game

Group: 1 Type: A	Group: 2 Type: B	Group: 3 Type: C	Group: 1 Type: D	Group: 2 Type: E
Group: 3 Type: F	Group: 1 Type: G	Group: 2 Type: A	Group: 3 Type: B	Group: 1 Type: C
Group: 2 Type: D	Group: 3 Type: E	Group: 1 Type: F	Group: 2 Type: G	Group: 3 Type: A
Group: 1 Type: B	Group: 2 Type: C	Group: 3 Type: D	Group: 1 Type: E	Group: 2 Type: F
Group: 3 Type: G	Group: 1 Type: A	Group: 2 Type: B	Group: 3 Type: C	Group: 1 Type: D
Group: 2 Type: E	Group: 3 Type: F	Group: 1 Type: G	Group: 2 Type: A	Group: 3 Type: B

Fig. 2. Thirty generic tiles generated, with the properties group, type, and colour, to use as a sample for the random rule generator.

and a request to generate a story that can contextually explain the rules without mentioning them explicitly.

OpenAIs ChatGPT 3.5-turbo API [8] is currently used to prompt for a story, unfortunately, the language model produces stories that are less relatable for children such as "Your boss asks you to..." and uses words that are not exactly child friendly.

V. CURRENT RESULTS

To test the procedural content generator combined with the genetic algorithm, repeated tests were conducted five hundred times with a fixed population size of one hundred rules and a mutation rate of fifty percent. For each test, a unique target between zero and all possible solution sets was chosen, and a maximum of fifty generations. The algorithm was able to approximate the target set of solutions with 99.9 percent on average with a standard deviation of 0.17 percent within an average of 22.3 generations and a standard deviation of 16.6 generations.

Given the constraining nature of the rules, thirty generic tiles were generated with three properties, shown in figure 2, each with a different value count (three, five, and seven). The three properties and values were *group*, which had three values (1,2,3), *colour* which had five values, and *type* which had seven values (A, B,..., G). These were sampled during initialization by the random generator and again during mutation, which allowed for more dynamic outcomes for the procedural rule generator.

The difficulty evaluation is the main bottleneck for the performance of the genetic algorithm as it uses brute force for all possible combinations on the entire population.

Here are three samples with different targeted solution sets to give a sense of the rules that the procedural generator outputs. For an almost trivial problem (141,966 solutions) the rules limit tiles which there are only a limited number of: "There can only be 3 with type set to A and There can only be 3 with colour set to red". The first rule only affects solutions where more than 3 of the 5 tiles with type A are present and

Prompt

This is a game where one or more players have to match exactly five tiles featuring an image each.

The players have the following images available:
 [Lion, Cheetah, Eagle, Hyena, Black mamba, Leopard, Zebra, Giraffe, Gazelle, Hornbill, Elephant, Ostrich, Tiger, Bear, Crane, Cobra, Crocodile, Komodo Dragon, Panda, Orangutan, Tapir, Peacock, Rhino, Red Panda, Falcon, Wolf, Viper, Bison, Red deer, Horse]

Only images that follow these rules will be accepted:
 "Those with classification set to mammal cannot be adjacent to those with preferred food set to meat and Those with family set to cat cannot be adjacent to those with family set to horse"

Create a brief story about the intention to host a dinner for the items featured in the images. The story should contain conflicts or agreements that comply with the rules and provide reasons for them. Avoid mentioning the images or the rules. End by asking the players to choose five tiles, after which the story will continue. Address the players as you, and limit the story to five sentences.

Output

In a land where unity and appreciation for nature thrived, a magnificent dinner was planned to honor the majestic creatures that roamed the realm. However, an inherent conflict surfaced as certain mammals clashed with their carnivorous tendencies, threatening to disrupt the serene ambiance. To preserve the harmony, a wise decree was enacted, decreeing that those from feline and equine families must remain apart, ensuring peace and respect among them. Now, dear players, the time has come for you to carefully select five extraordinary beings who will grace this momentous occasion. Choose wisely, for the symphony of the dinner shall depend on your choices.

Fig. 3. Prompt to generate a story from a game definition consisting of tiles and procedurally generated rules.

the second does the same for solutions where 3 of the 6 tiles with the colour red are present.

For a constrained problem (186 solutions), the rules tend to exclude many tiles by targeting a common property: "Exclusively for anywhere group is equal to 1". Since there are only three group values, this effectively excludes ten tiles entirely. Between these two extremes (70,764 solutions), the more complex rule concepts tend to be combined: "Those with group set to 3 cannot be adjacent to those with type set to E and Those with colour set to green cannot be adjacent to those with colour set to white" These rules partially exclude some tiles when other tiles are present in the solution which combined with common properties such as group and colour provide finer-graded control of the solution sets.

For the language model to provide a narrative context the tiles had to feature themed entities, which can be incorporated in a story and has a means to conflict. This resulted in a game where the players have to host a dinner for various animals, where individual animals have conflicting constraints that the player has to abide by. Before the large language model is utilized a prompt is compiled, shown in figure 3, consisting of a game definition where all tiles are mentioned by name followed by the procedurally generated rules and finally instructions to generate a story.

VI. FUTURE PLANS

In the next iteration of the project, we plan to train a language model specifically on children's literature, provided by the Gutenberg project². This should produce more children-themed stories and hopefully use a language that children are more familiar with.

²<https://www.gutenberg.org/>

Large language models have been used to guide reinforcement learning with the use of prompting techniques and evaluation of text output [4]. We will try to use large language models to analyze the communicated rules and use prompting techniques to get suggestions for possible solutions. These suggestions will then be used to guide a solver algorithm and at the same time determine comprehensive difficulty based on the accuracy of the predictions.

The current approach also lacks actual player feedback on difficulty, which will be an important factor to balance and guide future experiments. Players will be invited to play test the prototype, while data is collected to model their behaviour in response to different levels of difficulty.

REFERENCES

- [1] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym. 2016.
- [2] Jaime Carbonell. AI in CAI: An Artificial-Intelligence Approach to Computer-Assisted Instruction. *IEEE Transactions on Man Machine Systems*, 11(4):190–202, December 1970.
- [3] Lucrezia Crescenzi-Lanna. Multimodal Learning Analytics research with young children: A systematic review. *British Journal of Educational Technology*, 51(5):1485–1504, September 2020.
- [4] Yuqing Du, Olivia Watkins, Zihan Wang, Cédric Colas, Trevor Darrell, Pieter Abbeel, Abhishek Gupta, and Jacob Andreas. Guiding Pretraining in Reinforcement Learning with Large Language Models, February 2023. arXiv:2302.06692 [cs].
- [5] D. Hooshyar, M. Yousefi, M. Wang, and H. Lim. A data-driven procedural-content-generation approach for educational games. *Journal of Computer Assisted Learning*, 34(6):731–739, December 2018.
- [6] Nina Iten and Dominik Petko. Learning with serious games: Is fun playing the game a predictor of learning success? *British Journal of Educational Technology*, 47(1):151–163, January 2016.
- [7] Su Mu, Meng Cui, and Xiaodi Huang. Multimodal Data Fusion in Learning Analytics: A Systematic Review. *Sensors*, 20(23):6856, November 2020.
- [8] OpenAI. (no date provided). Introducing ChatGPT. <https://openai.com/blog/chatgpt> accessed 13. May 2023.
- [9] Joon Sung Park, Joseph C. O’Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative Agents: Interactive Simulacra of Human Behavior, April 2023. arXiv:2304.03442 [cs].
- [10] J.L Plass, B.D. Homer, S. Pawar, C. Brenner, and A.P. MacNamara. The effect of adaptive difficulty adjustment on the effectiveness of a game to develop executive function skills for learners of different ages. *Cognitive Development*, 49:56–67, January 2019.
- [11] Symeon Retalis. Creating Adaptive e-Learning Board Games for School Settings Using the ELG Environment. 2008.
- [12] Stefanie Vanbecelaere, Katrien Van Den Berghe, Frederik Cornillie, Delphine Sasanguie, Bert Reynvoet, and Fien Depaepe. The effectiveness of adaptive versus non-adaptive learning with digital educational games. *Journal of Computer Assisted Learning*, 36(4):502–513, August 2020.
- [13] Julie Villesseche, Olivier Le Bohec, Christophe Quaireau, Jeremie Nogués, Anne-Laure Besnard, Sandrine Oriez, Fanny De La Haye, Yvonnick Noel, and Karine Lavandier. Enhancing reading skills through adaptive e-learning. *Interactive Technology and Smart Education*, 16(1):2–17, March 2019.
- [14] Georgios N. Yannakakis and Julian Togelius. Experience-driven procedural content generation (Extended abstract). In *2015 International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 519–525, Xi’an, China, September 2015. IEEE.
- [15] Yu Zhonggen. A Meta-Analysis of Use of Serious Games in Education over a Decade. *International Journal of Computer Games Technology*, 2019:1–8, February 2019.