Hexagram Tangram Puzzle Artifact

Yazeed Mahmoud[1], Nathan R. Sturtevant[1,2]

[1]Computing Science Department, University of Alberta
[2]Alberta Machine Intelligence Institute (Amii)
yazeed1@ualberta.ca, nathanst@ualberta.ca

Abstract

This paper describes the artifact associated with an AIIDE 2024 paper on the use of AI tools for the creation of a Hexagram Tangram Puzzle. We describe our motivation for creating the puzzle, the layout and construction of the puzzle, and what is publicly available with the artifact.

Introduction

Game design can be a difficult and time-consuming process. In another paper (Mahmoud and Sturtevant 2024), we considered the process of puzzle design motivated by the puzzle shown in Figure 1(a). This is a hexagon tangram puzzle that we purchased and studied, but found the original design to be disappointing. The goal is to place all pieces onto the board without overlap. In the original design, the pieces have colors, but the colors serve no functional purpose. There are hundreds of solutions to the puzzle, but nothing to motivate finding or exploring the puzzle space. While we took personal initiative to study and understand the puzzle, we took it as a challenge to use AI tools to create a richer design than the original puzzle, following work in co-creative design (Liapis, Yannakakis, and Togelius 2013; Yannakakis, Liapis, and Alexopoulos 2014; Smith et al. 2012; Butler et al. 2013). This is a puzzle that can be represented both digitally and physically, so, given the initial motivation, we decided to focus on building a puzzle that could be realized both physically and digitally.

The full design process used to come up with the final puzzle and associated curriculum are described elsewhere (Mahmoud and Sturtevant 2024). The purpose of this paper is to describe the publicly available artifacts associated with our work. This includes the code that generated the puzzles, 3D models of the puzzle in `stl` format, all possible legal puzzle instances in `svg` format, and a curriculum of puzzle instances.

Puzzle Design

In this section we describe the final design of the puzzle. Our instance of the puzzle was designed through an iterative process of human design interleaved with Exhaustive Procedural Content Generation (EPCG) (Sturtevant and Ota 2018).

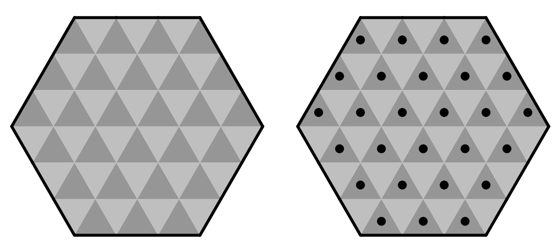
(a) (b)

Figure 1: (a) Reference puzzle and (b) near-final product

This process used EPCG as a system to answer design questions about the emerging design space of a Hexagon Tangram puzzle, starting from the choice of pieces in the puzzle, and ending with the final curriculum.

Board Design

The board used in the design, seen with pieces on it in Figure 1(b) is shown more abstractly in Figure 2(a) and (b). The hexagon shape is made up of 54 smaller triangles, where each piece will overlap three or six of these triangles. The two sides of the board are used to create a variety of constraints, and thus richer puzzle instances. One side is unconstrained, meaning that any piece can be placed in orientation and in any position without constraint. The second side of the board has bumps which physically limit which side of a piece can be physically placed on the board, as well as in



(a) (b)

Figure 2: (a) Hexagon puzzle board. The board is made up of 54 triangles; pieces are placed aligned to triangles. (b) Constraint locations.

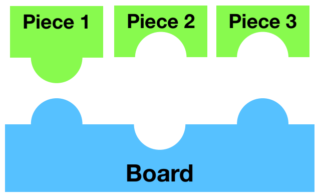


Figure 3: Every triangle on the board or on a piece has either a bump or a hole. Bumps cannot be placed opposite a bump (piece 1), but holes can be aligned with either a hole (piece 2) or a bump (piece 3).

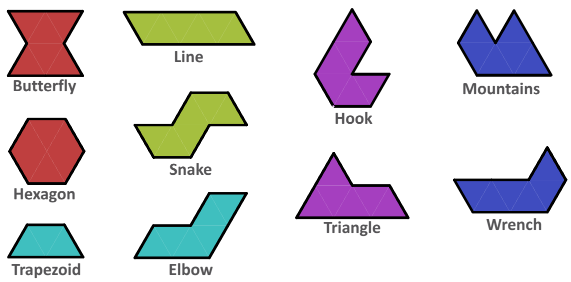


Figure 4: The selected set of pieces and the names assigned to them (excluding the "Trapezoid" piece). Notice how they are all composed of two trapezoids in some configuration.

which positions a piece can be placed.

Board constraints on piece placement are easy to implement digitally, but we wanted our constraints to be implementable both physically and digitally. The physical realization of the constraints is shown in Figure 3. Every triangle which is part of a piece and part of the board either has a bump or a hole. When a bump is present, it must be aligned with a hole, but holes can be aligned with both bumps and holes.

Piece Design

There are 10 different pieces in the puzzle. One of these pieces is a trapezoid, which is composed of three triangles. All other pieces are formed by merging two trapezoids to create a larger compound piece with six triangles. Because there are 54 triangles on the board, one piece type is omitted in every puzzle instance. If the trapezoid is used in a puzzle, two trapezoids are provided. The original puzzle has four trapezoids and does not include the snake or butterfly pieces, while our puzzle contains one of each possible piece type. The number of puzzle solutions possible when a given piece type is omitted is found in Table 1. There are only nine solutions that do not involve trapezoids, so we included trapezoids in order to increase the size of the puzzle space.

One of the features optimized in our design is how to place bumps on pieces and the board. We chose to have a bump on every other triangle on the board, and thus needed to do something similar on each of the pieces. Due to piece symmetry, not all pieces should have bumps on them. For instance, if the hexagon and butterfly can be placed in at least one location on the board, they will be able to be placed anywhere by either flipping them over or rotating them. Thus,

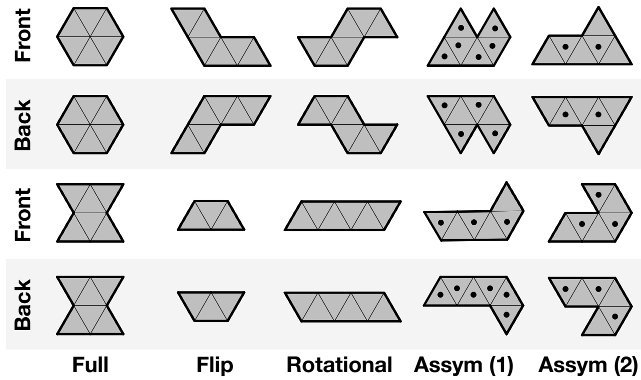


Figure 5: Final piece patterns. Holes are not drawn; bumps are black circles. Pieces are grouped by their symmetry patterns.

we only placed bumps on the four pieces that were asymmetric. The placement of bumps in the design is shown in Figure 3. We then colored the pieces according to their symmetry groups, as seen in Figure 4.

Our physical prototypes were created by 3D-printing the puzzles and using alcohol inks to paint the individual pieces. After this they were sealed with a clear coat of paint.

Curriculum

We built a curriculum focused around exploring constraints between pieces. There are two sections of the curriculum, one for each side of the board. In each of these sections there are chapters based on constraints around the colors of pieces. For instance, two pieces of the same color may be required to have touching edges, touching corners, or not be allowed to touch. Similarly constraints can be applied to pieces of different colors.

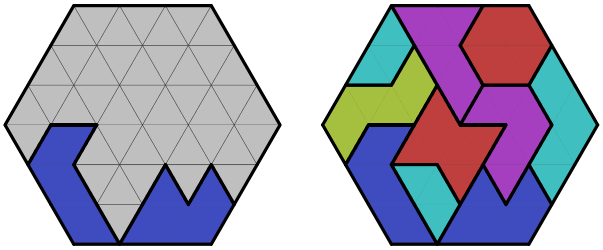
Chapters in the curriculum all have the same type of constraints, with 12 puzzles in each chapter. There is an additional introductory chapter that has no color constraints.

Given the solution to a puzzle, a puzzle instance can be created by removing pieces from the board to find a minimal set of pieces that can be uniquely solved. This is shown in Figure 6. We can optionally add more pieces onto the board to make a puzzle easier, which we do in the first chapter of the curriculum.

Within a chapter, puzzles are ordered according to their

Excluded	Solutions	Excluded	Solutions
Hexagon	673	Hook	179
Butterfly	352	Triangle	176
Wrench	321	Elbow	129
Snake	307	Line	97
Mountains	265	Trapezoid	9

Table 1: Number of solutions found with each possible piece excluded from the subset. There are 2,508 total solutions among all subsets.



(a) (b)

Figure 6: (a) A puzzle instance and (b) the unique solved board. (The line piece is disallowed in this puzzle)

entropy (Chen, White, and Sturtevant 2023). Put simply, the difficulty (entropy) of a puzzle is reduced when the current configuration of pieces on the board leads to situations where (1) only a single piece can be placed in a given location or (2) a given piece only has the possibility of being placed in a single location.

Artifact

The artifact described here is available from https://movingai.com/hextan.html. The artifact page includes:

- **Code** that uses EPCG to generate all possible puzzles solutions and corresponding starting states. This code is already publicly available under a MIT license from the HOG2 repository (https://github.com/nathansttt/hog2/tree/PDB-refactor/apps/hexagon), but may eventually be moved out of that respository. An extracted snapshop of the code from the current project is provided. A simple `build` script is provided that will compile the code using `g++`. By default, the code just runs the EPCG process and generates a curriculum.
- **Models** of the resulting puzzle and board. These models are generated from a geometric description of the pieces which can be exported from the code. We have used high resolution voxel models for 3D printing (0.05mm per voxel, so the print is at the same resolution as our 3D printer). The files used for printing are too large to include currently, but we are continuing to work on reducing file sizes, and may provide higher resolution files in the future.
- **Puzzles** from the curriculum. These puzzles are generated by running the provided code, but the output is given to avoid the need to run the code. The first output (`all_puzzles`) is the full set of 2508 puzzles that result from the puzzle design. Symmetric puzzles are removed. These puzzles are then analyzed and divided into two sets of six directories. The first directory (`gen_puzzles`) contains the solutions, while the second (`gen_puzzles_init`) contains starting positions for the curriculum. Puzzles are named by the entropy of the puzzle followed by the puzzle number.
- A **curriculum** of puzzles taken from the generated set of puzzles.

While several interactive prototypes have been developed during the span of this project, they are not in the current code since the physical prototype is sufficient for testing, and the prototypes were not maintained as the design evolved.

Conclusions

This paper has described the artifacts associated with the design of a Hexagon Tangram puzzle (Mahmoud and Sturtevant 2024). The puzzles, their 3D models, and the curriculum are now available for non-commercial usage. The code for the project is expected to continue to receive updates as we consider refinements to our design.

Acknowledgements

Augustin Blanchonnet developed the first prototype of this work, exploring possible puzzles and curriculum. This project was later re-written from scratch in C++ by the authors of this paper.

This work was supported by the National Science and Engineering Research Council of Canada Discovery Grant Program and the Canada CIFAR AI Chairs Program.

References

Butler, E.; Smith, A. M.; Liu, Y.-E.; and Popovic, Z. 2013. A mixed-initiative tool for designing level progressions in games. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13, 377–386. New York, NY, USA: Association for Computing Machinery. ISBN 9781450322683.

Chen, E. Y. C.; White, A.; and Sturtevant, N. R. 2023. Entropy as a Measure of Puzzle Difficulty. In *Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*.

Liapis, A.; Yannakakis, G. N.; and Togelius, J. 2013. Sentient sketchbook: computer-assisted game level authoring. In *Foundations of Digital Games*. ACM.

Mahmoud, Y.; and Sturtevant, N. R. 2024. Using EPCG for Designing a Hexagon Tangram Puzzle. In *Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*.

Smith, A. M.; Andersen, E.; Mateas, M.; and Popović, Z. 2012. A case study of expressively constrainable level design automation tools for a puzzle game. In *Proceedings of the International Conference on the Foundations of Digital Games*, FDG '12, 156–163. New York, NY, USA: Association for Computing Machinery. ISBN 9781450313339.

Sturtevant, N.; and Ota, M. 2018. Exhaustive and semi-exhaustive procedural content generation. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 14, 109–115.

Yannakakis, G. N.; Liapis, A.; and Alexopoulos, C. 2014. Mixed-initiative co-creativity. Foundations of Digital Games.