My Father and Martin Gardner

Lyman Hurd - G4G16, 2026

Introduction

I was introduced to Martin Gardner's columns at an early age by my father, John Coolidge Hurd. A notable theologian [1], he received his bachelor's degree from Harvard in chemistry and never lost interest in the sciences. He and I would eagerly await each month's Scientific American and invariably the first thing we turned to was the Mathematical Games column (to be completely honest there were months in which that was the only article we read in its entirety). When the collections of Martin's columns started appearing, it was understood what my Christmas present to him would be.

He passed away this past October at the age of 97 and I have wanted to present this paper for quite some time and I have been spurred to completion. After a great deal of editing, I settled on three somewhat different examples of columns. Their importance is mainly personal as they represent projects undertaken with him that still bring back memories. I am hoping that the reader will excuse the somewhat stream of consciousness presentation. In their own way, each of these presents my personal madeleine.

References

[1] **"In Memoriam: John Coolidge Hurd,"** *Trinity College*. Accessed Feb. 15, 2026. [Online]. Available:
`https://www.trinity.utoronto.ca/discover/news/item/in-memoriam-john-coolidge-hurd/`

Conway's Game of Life and APL

John Conway's Game of Life was introduced to the public by Martin Gardner's column [1] and the follow-up article [2]. The rules are well known. A grid is populated by white and black cells and in each iteration one looks at the number of black cells in the eight neighboring cells and if the total is 2 or 3 a cell that is occupied stays occupied, if an unoccupied cell has exactly 3 live neighbors, it becomes occupied and if an occupied cell has fewer than 2 or more than 3 occupied neighbors, it becomes unoccupied. Life has been shown to be as powerful as a universal Turing Machine [3].

One of my father's major academic accomplishments was the development of a system for teaching his divinity students New Testament Greek [4] written in the programming language APL [5].

We programmed this language on an IBM 360 owned by the University of Toronto accessed via a dial-up modem using an IBM 2741 terminal.

Figure 1: IBM360 [6]

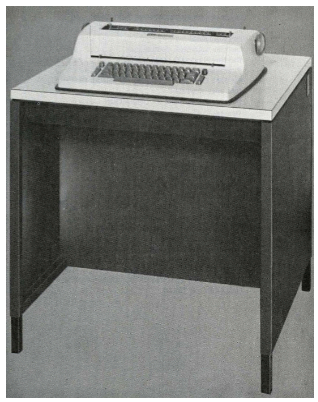

Figure 2: IBM 2741 Terminal [7]

Figure 3: An acoustic coupler [8]

Figure 4: An APL typeball [9]

Connection to the computer was managed via modem using an acoustic coupler. The connection was at a blazing 110 baud (bits/sec).

The terminal was effectively a smart version of the ubiquitous IBM Selectric typewriter which used a typeball enabling it to change character set by replacing the ball.

APL was known for its obscure character set. I figured that my father had a leg up on this because a number of the APL operators were cribbed from Greek. Many other characters I would eventually be quite familiar with as a working mathematician, but at this point in my life (i.e., middle school) that lay in the future. Here is the standard layout which at one point I used to have memorized.

Figure 5: The layout of a standard APL keyboard [10]

My father programmed a version of the game of life in APL and although his version is long gone, I attempted to reproduce what he had done using a modern implementation of APL [11]. Alongside some helper methods that convert between arrays of characters and numbers and which manage the iteration of steps, there is also a function PAD which determines if the evolution is about to exceed the bounds of the provided box and extends each of the sides as needed (note that as written PAD can only make the bounding box bigger and does not attempt to remove space, therefore a glider structure would be represented by bigger and bigger grids).

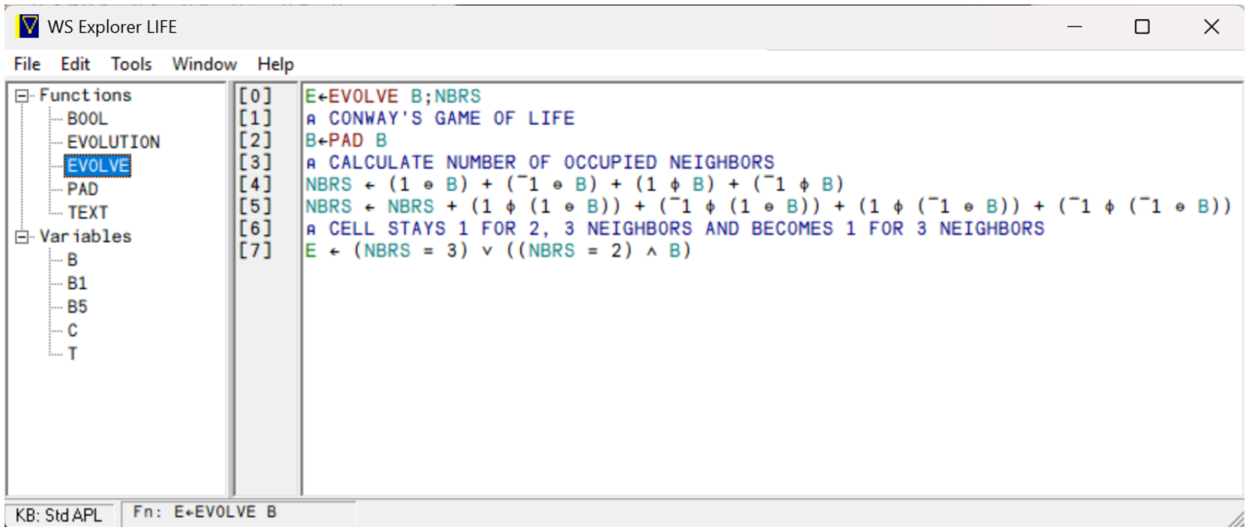

Figure 6: An APL implementation of the evolution rule in Life.

The calculation relies heavily on the fact that APL allows entire arrays to be acted on by a single operator as well as built-in operators that rotate rows and columns by a fixed amount. Note that APL recognizes the prefix of negative numbers as being distinct from the negation operator hence the representation of −1 in the program.

As an example, I show the evolution of the "R" pentomino [12] which has the distinction of being the most complex dynamics of any initial configuration with five or fewer live sites. Here is an example of the output of the program at 1, 10 and 100 steps. The evolution does not become stable until 1103, however space constraints prevented continuing the evolution that far.

```
STEP 1

 ___
| **|
|** |
| * |
 ___
```

```
STEP 10
 ________
|    *   |
|  *  *  |
| *      |
| *    * |
| **     |
|        |
|   ***  |
|        |
 ________
```

```
                               STEP 100


                                             **
                                            *  *
                                    **      *  *
                                    **       **
                        **
                       * *      **              *
     ***                 *      **          *****
    *  *                                        *
    *   *                                   ** ** **
   **  *                            **         *  *
     *                      **      ** **       **
      **                    **     *   **   ***
      ***                           **   *  ***
     ****                      ***  ** **
                                       **

     * *
    ** **
   * * **      **
  ** ****      **
    *****
   **  *
    ***
     *
```

Figure 7: Evolution of my Game of Life program for 1, 10, and 100 steps.

APL taught me several lessons about programming I had to unlearn as I discovered how algorithmic complexity actually works. There was a tendency to equate the complexity of a program with the number of symbols used to express it which was a dangerous assumption for a language which, for example, can invert a fully general matrix using a single symbol. There were also mechanisms for chaining together operations even if they were not logically connected and I used to be of the opinion that anyone writing a program of more than one line was not trying hard enough. I once taught a student linear algebra and he told me that the previous summer he had worked as an intern at IBM and part of his job was inverting large orthogonal matrices using APL. I said, "oh, you mean the transpose" and he said that as he had not taken my class yet, he used the APL operator "domino" as it and the transpose are represented by the same number of symbols in APL.

Another side note on this section is that years later I defended my dissertation at Princeton which concerned cellular automata and due to a scheduling conflict, my advisor, John Milnor, could not be there and John Conway volunteered to chair my defense committee.

References

[1] M. Gardner, “The fantastic combinations of John Conway’s new solitaire game ‘Life,’” *Scientific American*, vol. 223, no. 4, pp. 120–123, Oct. 1970.

[2] M. Gardner, “On cellular automata, self-reproduction, the Garden of Eden and the game ‘Life,’” *Scientific American*, vol. 224, no. 2, pp. 112–117, Feb. 1971.

[3] “Conway’s Game of Life,” *Wikipedia*. Accessed Jan. 10, 2026. [Online]. Available: https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life (en.wikipedia.org in Bing)

[4] “Koine Greek,” *Wikipedia*. Accessed Jan. 10, 2026. [Online]. Available: https://en.wikipedia.org/wiki/Koine_Greek (en.wikipedia.org in Bing)

[5] L. Gillman, *APL/360: An Interactive Approach*. New York, NY: John Wiley & Sons, 1970.

[6] S. Loftesness, “IBM System 360 Model 30 – Computer History Museum – 2014,” *Flickr*. Accessed Jan. 10, 2026. [Online]. Available: https://www.flickr.com/photos/scottloftesness/13437981955/ (flickr.com in Bing)

[7] “A24-3415-2_2741,” *Sharktastica*. Accessed Jan. 10, 2026. [Online]. Available: https://sharktastica.co.uk/resources/images/typewriters/A24-3415-2_2741.jpg (sharktastica.co.uk in Bing)

[8] “AcousticCouplerAK2000Tel,” *Wikimedia Commons*. Accessed Jan. 10, 2026. [Online]. Available: https://commons.wikimedia.org/wiki/File:AcousticCouplerAK2000Tel.jpg (commons.wikimedia.org in Bing)

[9] “APLTypeBall,” *APL Wiki*. Accessed Jan. 10, 2026. [Online]. Available: https://aplwiki.com/images/4/48/APLTypeBall.jpg (aplwiki.com in Bing)

[10] “APL-keybd2,” *Wikipedia*. Accessed Jan. 10, 2026. [Online]. Available: https://en.wikipedia.org/wiki/Digital_encoding_of_APL_symbols#/media/File:APL-keybd2.svg (en.wikipedia.org in Bing)

[11] “APLX,” *MicroAPL*. Accessed Jan. 10, 2026. [Online]. Available: https://microapl.com/APL/index.html

[12] “R-pentomino,” *LifeWiki*. Accessed Jan. 10, 2026. [Online]. Available: https://conwaylife.com/wiki/R-pentomino (conwaylife.com in Bing)

The Binary System

The first chapter of Martin's book *New Mathematical Diversions from Scientific American* [1] gave an example of how to construct a computer based on indexed cards with notches strategically arranged to represent the numbers from 0-31 in binary, The notches in the card can be interpreted as 1's and the holes as 0's, and the numbers expressed as most to least significant bit, i.e., the first card would represent the number $01000_2 = 8$, and so on. By inserting a pencil in the leftmost hole and shaking, one could separate the cards with a 0 in a given position from those with a 1 while maintaining the order and by successively doing this to each position and then stacking the "1" cards behind the "0" cards, one can sort the cards independent of their original order.

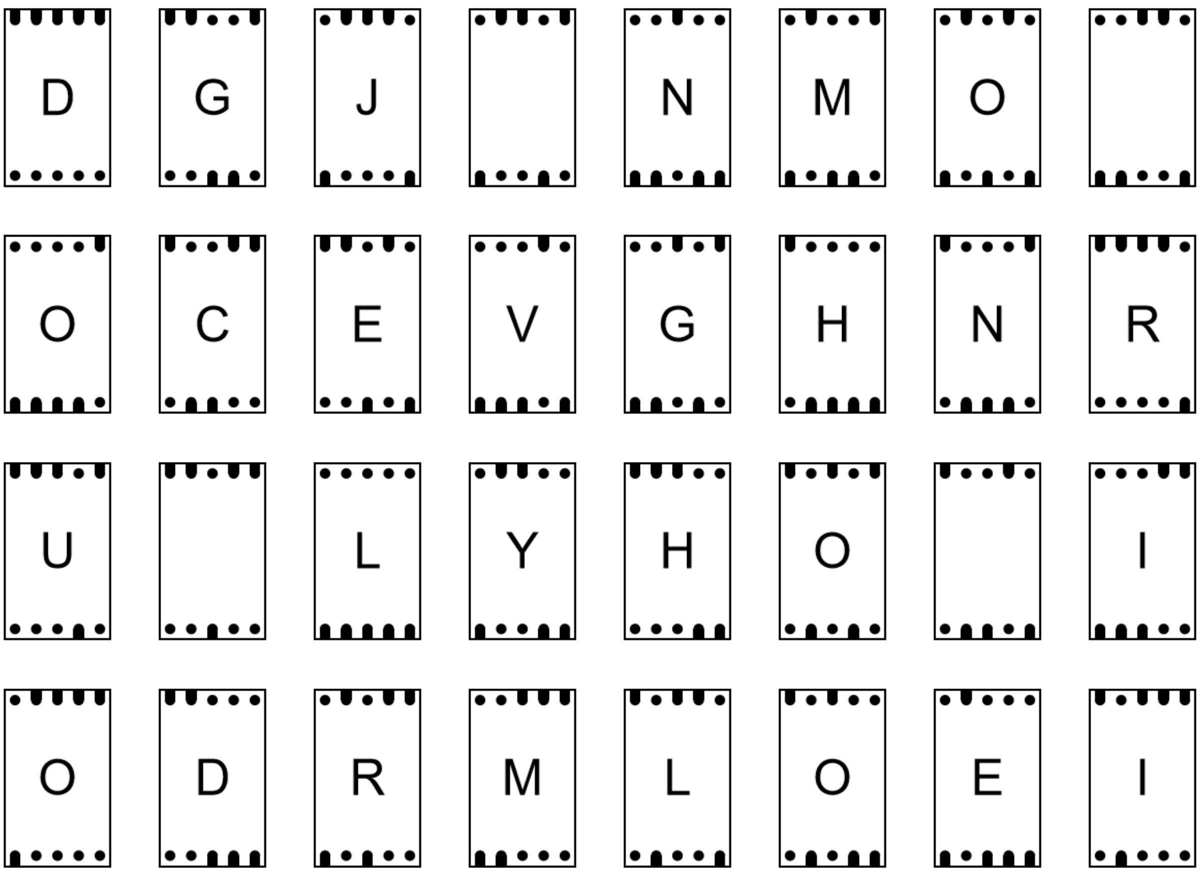

Figure 1: 32 sortable cards with a personal message

Later, the article describes how to approach logic problems for which each of the place values represents a proposition and one wishes to find a card that satisfies all the given conditions.

Always an optimizer, my father eventually introduced a complimentary set of holes on the bottom of each card which had a slot if and only if the corresponding column on top had a hole and vice versa. This enables one to use two pencils to separate the cards deterministically without resorting to shaking. Ever the perfectionist, he also devised handles from coat hangers (always a favorite source of building material) that looked more or less like the hooks pictured in the article.

To sort one starts at the rightmost column and pulling apart the cards puts the cards in the lower set of cards (those with a cutout) behind the cards in the upper pile. Proceed to do the same until one reaches the leftmost column. This is referred to as an "least significant but" ordering. As was the case in the illustration from Martin's book, I have encoded a message of my own which can be read from the cards in the correct order.

References

[1] M. Gardner, “The binary system,” in *New Mathematical Diversions from Scientific American*. Chicago, IL: Univ. of Chicago Press, 1966, pp. 1–13.

MacMahon's Colored Squares

My father had a great facility with tools of all descriptions and many of Martin's columns that dealt with physical objects, he was all over. We played with flexagons, he made us a set of tangrams, and most impressively he constructed a set of Penrose tiles for us out of wood. This last feat was especially impressive as it was done almost entirely with hand tools which was especially difficult in constructing the non-convex "dart" tiles.

I was always on top of the logical side of puzzles, but never was able to reproduce his real world handiwork. He also had a penchant for programming and that he passed on to me.

Martin Gardner wrote a series of articles [2]–[5] about various tiling puzzles introduced by the combinatorist Major Percy Alexander MacMahon [1]. This was exactly the sort of project my father enjoyed playing with and we made a set of handmade squares and triangles, and at one point we did improvise a set of cubes although I recall those being less immediately successful as I believe we glued on the colored labels instead of using paint.

When I described this paper to him in the early fall, I also sent him a copy of Kate Jones' commercial realization of the colored squares, Multimatch [6] (in fact I bought a plastic copy for him and a paper copy for myself).

Here are MacMahon's 24 tiles consisting of all possible 4-tuples of 3 colors, considering two tiles to be equivalent if they can be rotated into each other but *not* reflected.

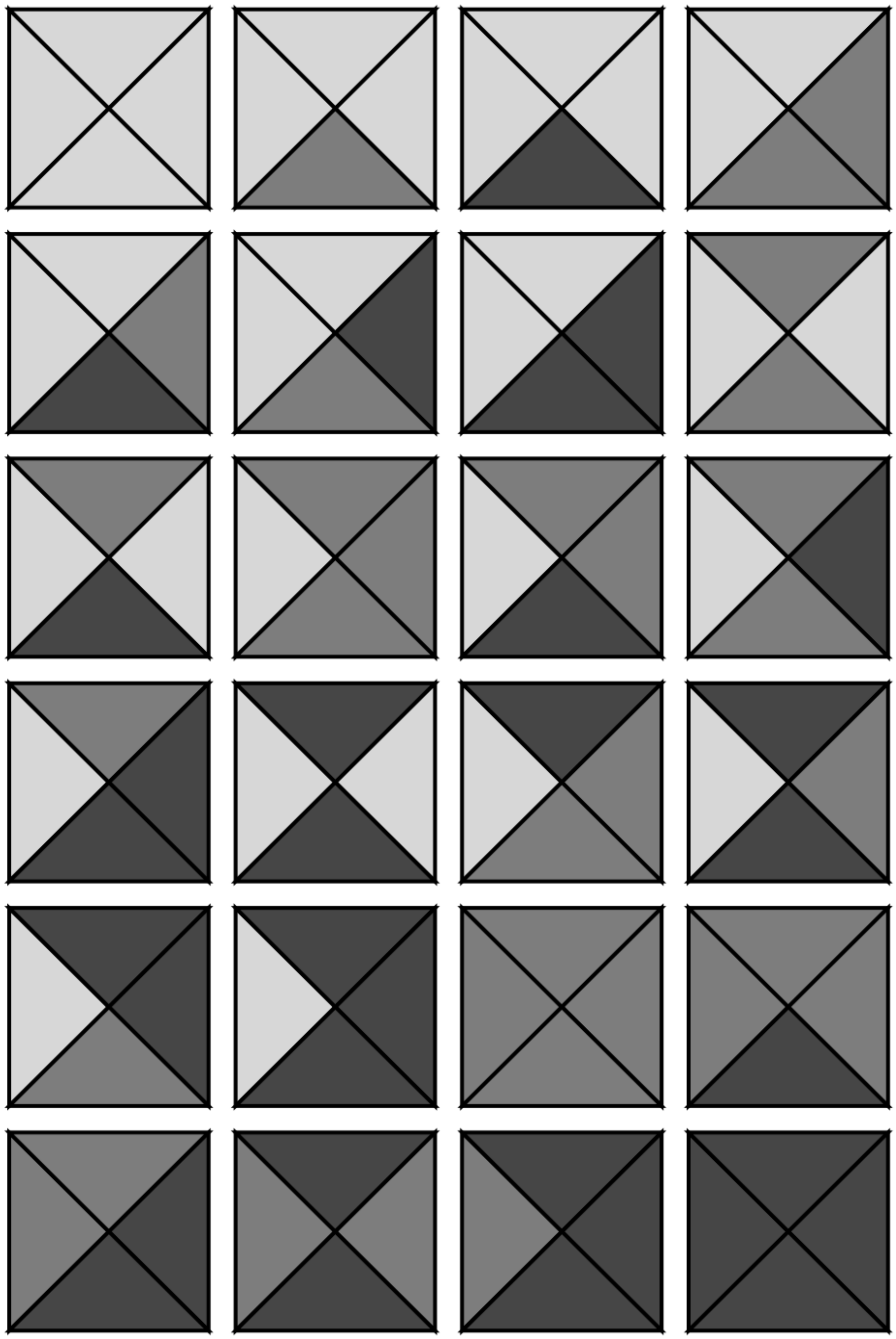

Figure 1: The 24 MacMahon Colored Squares

For each of these sections, I considered the best form of illustration and in this case I thought an appropriate demonstration would be to find all possible solutions of the 4 × 6 square with edges matched. In [7] they make the assertion that "In 1964 a super computer spent 40 hours to produce a list of 12,261 solutions to this puzzle." and I wanted to see if I could reproduce this result (in fact this number comes up short).

In 2016, in an attempt to write a paper for a previous G4G, I implemented Donald Knuth's Dancing Links algorithm from his original paper [8] in Python and have adapted that original program several times to solve problems of this sort. The original project concerned a specified set of polyominoes which my co-author, Jay Schindler, and I called "Euler Tiles", but unfortunately there did not seem to be as many interesting shapes tileable from this subset as originally hoped.

Since that work was done, I have subsequently read his modifications in Volume 5 of *The Art of Computer Programming* [9] which includes some optimizations of which I was previously unaware, and I believe that in my original attempt I reinvented the wheel in several instances, however for this project I worked with the code as I originally wrote it ten years ago.

The general approach is to solve the open cover problem, given a matrix whose rows consist of zeroes and ones, choose a subset of the rows such that every column has exactly one entry with a one. There are twenty-four columns allocated to indicate which tile is being described and therefore the fact that these columns each end up with a single occurrence of a one corresponds to the constraint that every tile is used exactly once. A further twenty-four columns represent locations and therefore every piece will have at least twenty-four rows that represent all possible locations. Again the column constraint indicates that every location can only accommodate a single tile. Note that there is a row for every possible orientation which may be as few as one for tiles that have a single tile or as many as four. Now the remaining columns indicate which color is on each side but with a slight twist. A specific color is encoded in one direction as, for example, (1, 0, 0) while if it appears on the other side of the associated edge as (0, 1, 1).

The time taken to retrieve all 13,328 solutions (after removing symmetric positions) was less than a day on my Lenovo Thinkpad laptop. Apparently the previous search had missed some, but without access to their solutions it is impossible to tell where the discrepancy arises especially as the two numbers do not differ by a multiple which would suggest that one approach took into account symmetries that the other did not.

Here is one such solution from my program.

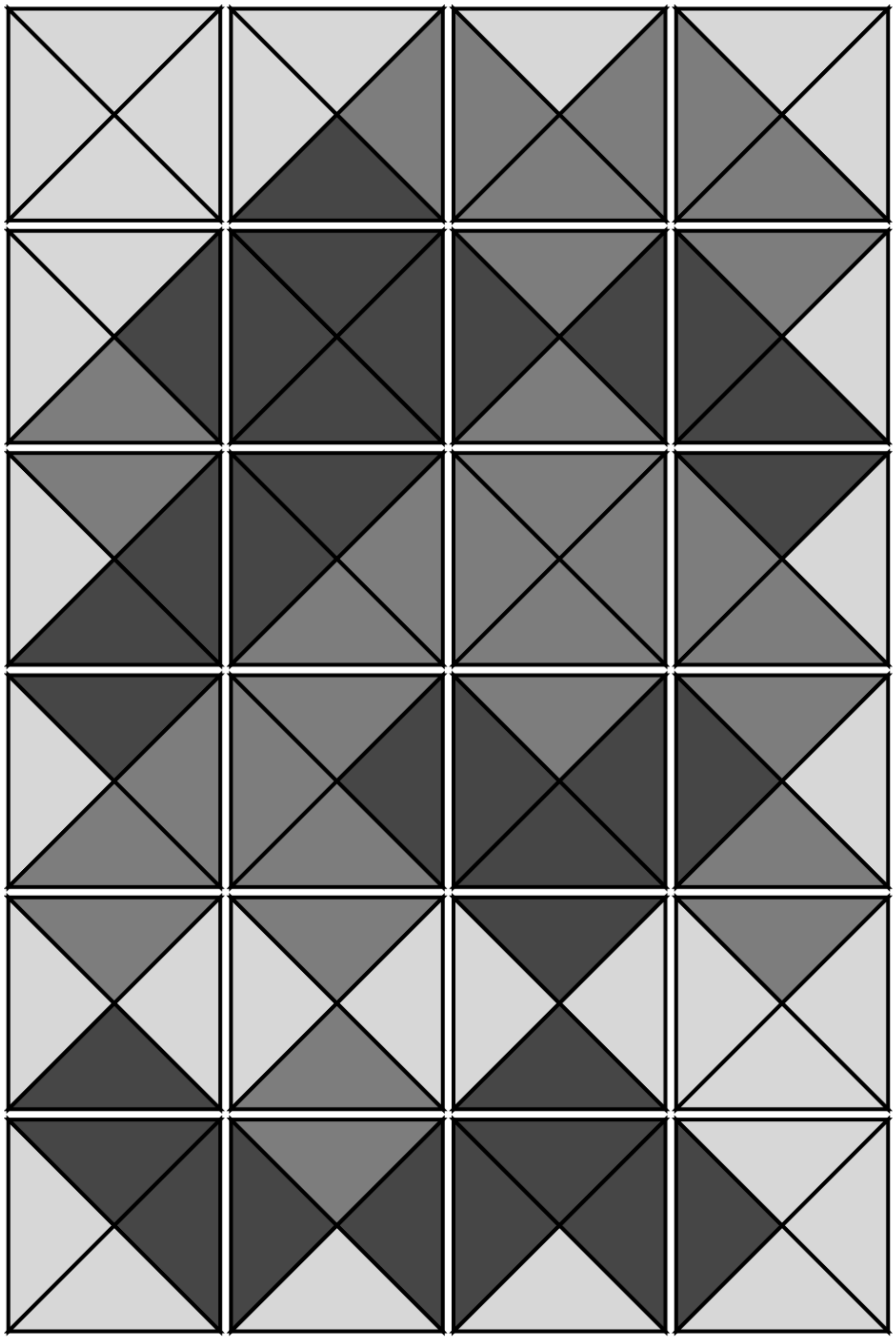

Figure 2: A candidate solution.

To reduce the search space we accounted for the following symmetries:

- permuting the three colors which (a factor of 6)
- reflecting either horizontally or vertically (a factor of 4)

Thus, one can, with care, reduce the search space by a factor of 24. In practice we achieve these savings by making the following assumptions:

- the boundary color must be light gray (factor of 3)
- the all gray square must be in the upper left quadrant (factor of 4)

This leaves just one remaining factor of 2 arising from permuting mid-gray and dark gray, the two non-boundary colors. To handle this we restricted the tile:

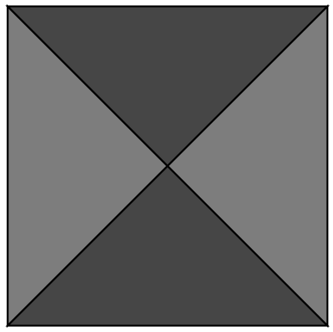

Figure 3: Tile 22

to only appear in this orientation, as any tiling that had it occur in the opposite orientation (i.e., rotated 90 degrees) could be replaced by a tiling with this orientation by switching all gray and dark gray triangles.

References

[1] R. A. Beeler, Major MacMahon’s Revenge. East Tennessee State University, n.d. Accessed Jan. 2, 2026. Available:
https://faculty.etsu.edu/beelerr/macmahon-revenge.pdf

[6] Kadon Enterprises, “Edgematching Colors and Shapes,” Gamepuzzles.com, n.d. Accessed Jan. 2, 2026. Available:
https://www.gamepuzzles.com/edgemtch.htm#M1

[7] “MacMahon 3-Color Squares,” Math on the McKenzie. Accessed Jan. 2, 2026. Available:
https://mathonthemckenzie.blogspot.com/2011/10/macmahon-3-color-squares.html

[8] D. E. Knuth, “Dancing Links,” Millennial Perspectives in Computer Science, 2000. Available:
https://arxiv.org/abs/cs/0011047

[10] “List of Martin Gardner Mathematical Games Columns,” Wikipedia. Accessed Jan. 2, 2026. Available:
https://en.wikipedia.org/wiki/List_of_Martin_Gardner_Mathematical_Games_columns#cite_note-math.upenn.edu/~kazdan/210S19/-5

[11] L. Hurd, euler-tiles. GitHub. Accessed Jan. 12, 2026. Available:
https://github.com/lymanhurd/euler-tiles/tree/master

Afterword

For those who have made it this far, I thank you for indulging me in this journey. I realized I had to edit my choices as when I first proposed this project, I was flooded with ideas. In middle school my seventh and eighth grade science fair projects were based on Martin's columns, one on topology and one on the Platonic and Archimedean solids. These studies influenced me to take up topology in earnest when I became a math major at MIT.

It is with some solemnity that I reflect that this will be the first time I will be attending G4G without my mandatory recap for my father of the things I have learned and the friends I made. I take some solace in the fact that I was able to share some early drafts of parts of this paper with him before he passed.