Game of Life on the Penrose Tiling

Examples of Still Lifes in Conway's Game of Life on the Kite-Dart Tiling

Zheng Fang
Advisor: Dr. May Mei
Department of Mathematics, Denison University

Introduction

Martin Gardner's article discusses still lifes in Conway's Game of Life. We can get these configurations by repeatedly running the three rules of game of life. [3] These configurations are all on the square tiling. Therefore, to explore whether this still life can have similarities in different tiling, we need to use Penrose kite-dart tiling, which is different from square tiling, and try to use the same game of life rules to find and more still lives.

We investigate the Game of Life on the Kite-Dart tiling.

Penrose Tiling

There are two different tiles in Penrose tiling. We call them the kite and the dart. There is a proportional relationship between the side length of the long side and the side length of the short side. This ratio is $\phi = \frac{1+\sqrt{5}}{2} \approx 1.6180$. [2]

Kite and Dart Substitution

Figures 1 and 2 show how to substitute a kite tile, and Figures 3 and 4 show how to substitute a dart tile.

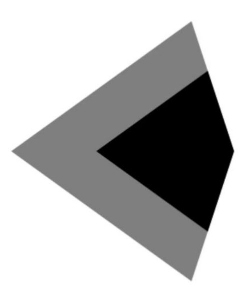

Figure 1. Enlarge the kite

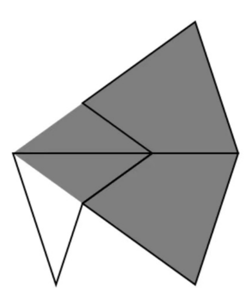

Figure 2. Substitute the kite by original size kites and dart

Figure 3. Enlarge the dart

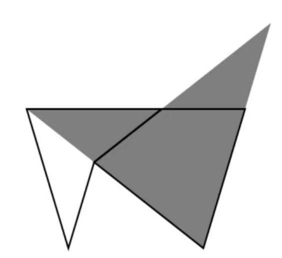

Figure 4. Substitute the dart by original size kite and dart

Based on an existing tiles, we can get a larger tiling through substitution. Every time we carry out substitution, we will first enlarge the tile in the patch according to a certain proportion, such as ϕ in Penrose tiling. We call this process one step of generation. The following Figure 4 shows our tiles at beginning. Because there are two different tiles as the basic graphics in the Penrose tiling, we implement two corresponding functions to draw the pattern. In the end, we have a Penrose tiling that large enough for next step.

Figure 5. The tiles we start with

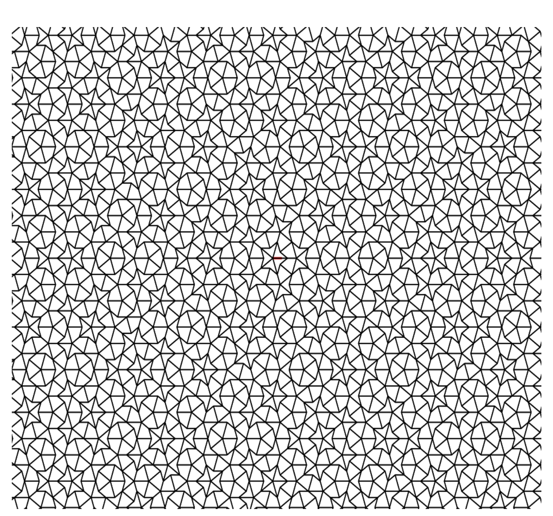

Figure 6. A patch from level 8 substitution

Neighbors for Each Tile in Kite and Dart Tiling

In order to find the neighbors of each tile and store them in a corresponding dictionary, we need to first determine the coordinates of the vertices of each tile to represent these positions. Then, the shared vertices in these coordinates are searched to determine whether the two tiles are adjacent tiles by edge or just a vertex.

Conway's Game of Life

In the Conway's Game of Life, there is an infinite orthogonal grid of square cells, each of them is in one of two possible states, alive or dead. At the same time, every cell only interacts with its 8 neighbours, which means the cells that are horizontal, vertical, or diagonal neighbours. Based on this neighbourhood, we can apply the following three rules to make the next generation. Births and deaths occur simultaneously, and the rules also continue to be applied repeatedly to create further generations. The three rules of survival, death, and birth are:

Survival: Every counter with two or three neighboring counters survives for the next generation.

Death: Each counter with four or more neighbors dies (is removed) from overpopulation. Every counter with one neighbor or none dies from isolation.

Birth: Each empty cell adjacent to exactly three neighbors-no more, no fewer-is a birth cell. A counter is placed on it at the next move.[3]

The Game of life on Penrose Tiling

We use the jld2 and fileio packages to store information from different notebooks, such as a dictionary that stores neighborhood relationships. In this way, we can directly access the calculated data in other files. Through the random function, we can get a random number list, which is used to randomly generate a list containing some tiles. We distinguish lists of alive tiles by find_black_tile function and dead tiles by find_white_tile function. find_neighbor_of_black function can help us find the number of surviving neighbors corresponding to each alive tile. Also, find_neighbor_of_white function can help us find the number of surviving neighbors corresponding to each dead tile. Finally, run_rules_of_GoL function is used to implement the transformation rules of Game of Life.

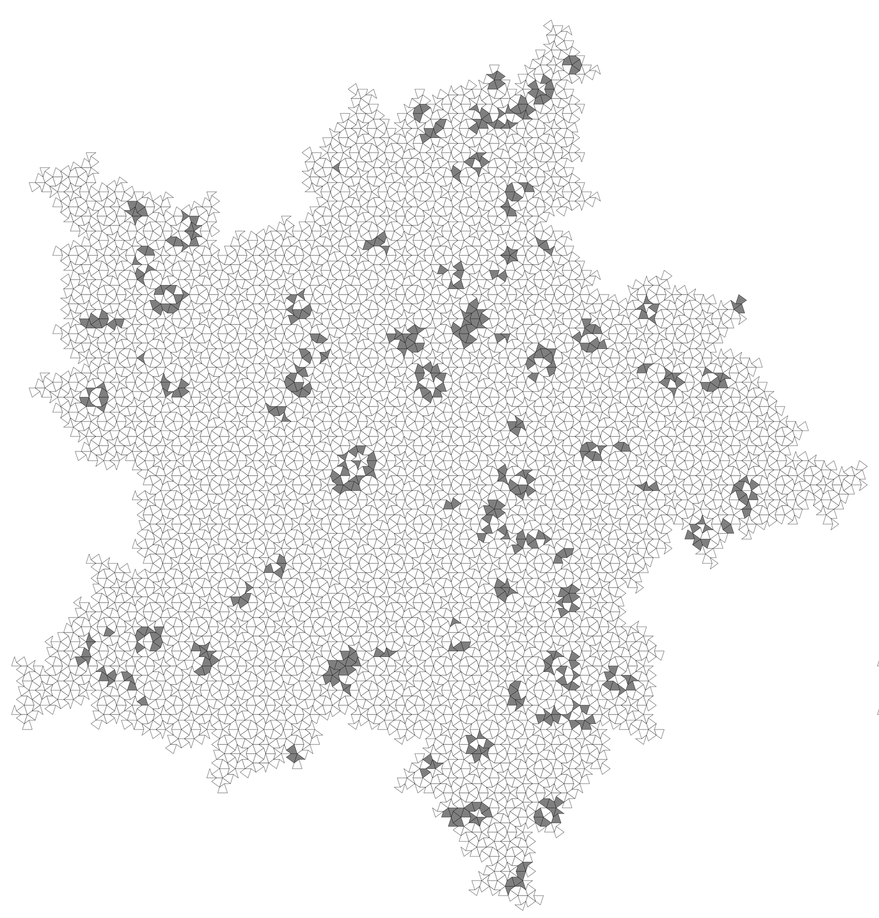

Figure 7. The first generation by Game of Life

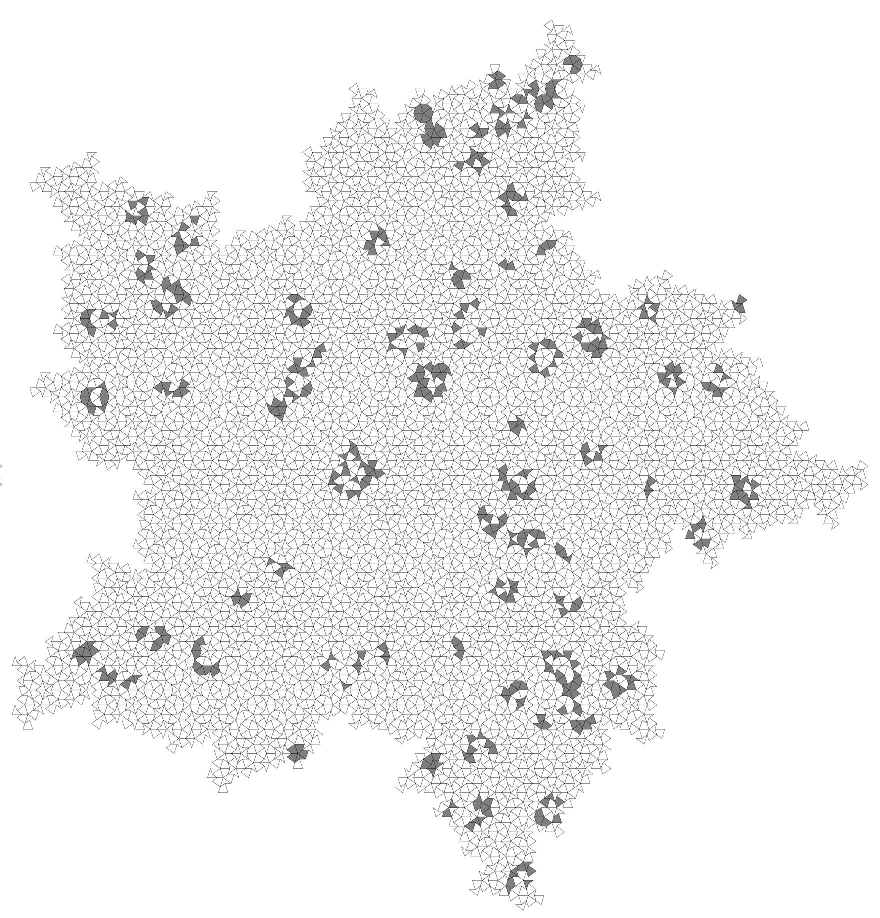

Figure 8. The second generation by Game of Life

Examples of Still Lifes with 3,4,5, or 6 Tiles

3-tiles still life

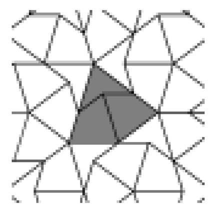

Figure 9.

4-tiles still lifes

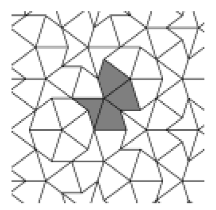

Figure 10.

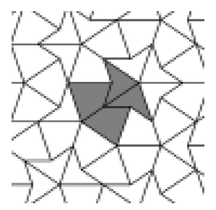

Figure 11.

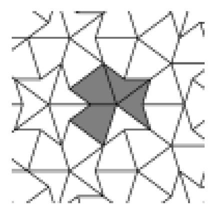

Figure 12.

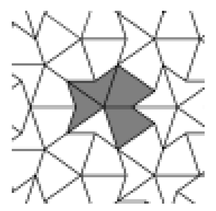

Figure 13.

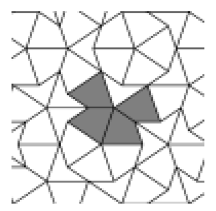

Figure 14.

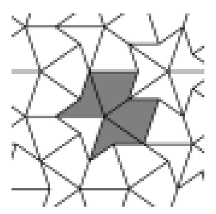

Figure 15.

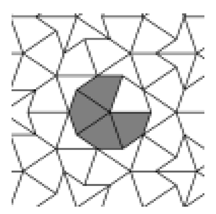

Figure 16.

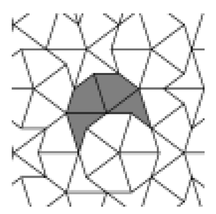

Figure 17.

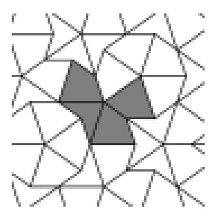

Figure 18.

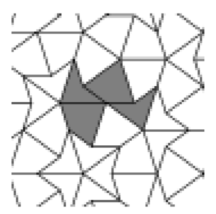

Figure 19.

5-tiles still lifes

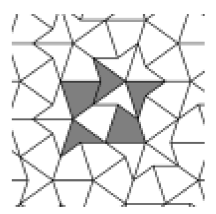

Figure 20.

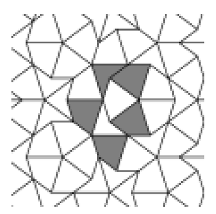

Figure 21.

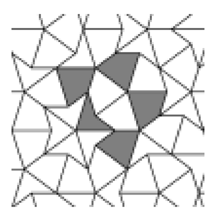

Figure 22.

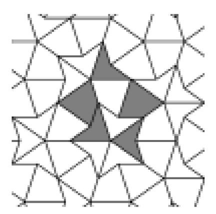

Figure 23.

6-tiles still life

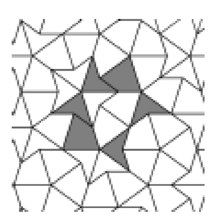

Figure 24.

Acknowledgement

We thank J. Reid and Polly Anderson Endowed for funding our research this summer. At the same time, we want to thank Professor Mei May for her support and encouragement, and Raghav and Mandy for their hard work and help. We would also like to thank Anish Athalye for providing this poster template.[1]

References

[1] Anish Athalye. "Gemini theme". In: *https://github.com/anishathalye/gemini* ().

[2] Martin Gardner. "MATHEMATICAL GAMES, Extraordinary nonperiodic tiling that enriches the theory of tiles". In: *Scientific American* 236.1 (1977), pp. 110–121.

[3] Martin Gardner. "MATHEMATICAL GAMES, the fantastic combinations of John Conway's new solitaire game life". In: *Scientific American* 223.4 (1970), pp. 120–123.