The Game of Life on Penrose Tilings: Robinson Triangle

Seung Hyeon Mandy Hong
Department of Mathematics, Denison University
Research Advisor: Dr. May Mei

Introduction

John Horton Conway's *Game of Life* gained attention in different areas because of the concept that it generates complexity from simplicity. [1] If we are given three simple rules: Survival, Death, and Birth, and an initial configuration, we can play the *Game of Life*. Compared to the simple setting of this game, we get surprising amount of complexity. Generally, *Game of Life* is played on the regular square tiling, which is a periodic tiling. However, we implement *Game of Life* on a nonperiodic tiling, specifically on the Robinson Triangle tiling, a variation of the Penrose tiling.

Game of Life

We call the square a *cell*. To play *Game of Life,* we start with a simple configuration of live cells on the square tiling and apply Conway's *genetic laws* to cells. [2]

Definition

Genetic laws apply three rules: Survival, Death, and Birth to cells.

1. Survival: Every cell with two or three live neighbors survives to the next generation.
2. Death: Each cell with four or more live neighbors or one or none dies.
3. Birth: A dead cell with three adjacent live neighbors is alive at the next generation.

Births and deaths co-occur, constituting a single generation. Applying the above rules to create subsequent generations, we find the population undergoing changes: Society could vanish, reach a stable configuration (still life), or repeat forever (oscillator).

Penrose Tiling

Definition

Suppose one can outline a region that tiles the plane by translation. We refer to such a tiling as *periodic.* A *nonperiodic tiling* is not periodic.

Among various nonperiodic tilings, one of the most well-known is the Penrose tiling, discovered by British mathematical physicist Roger Penrose. [3] In this research, we are particularly interested in the Robinson Triangle tiling, the variation suggested by the American mathematician Raphael M. Robinson. The four prototiles of the Robinson Triangle tiling is shown in Figure 1.

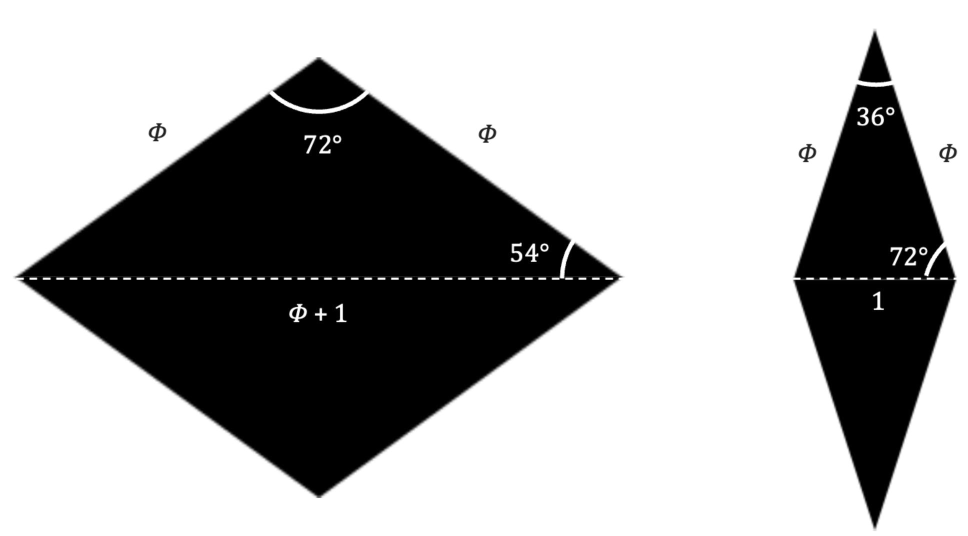

Figure 1: Four triangles are prototiles of the Robinson Triangle tiling.

Algorithms

To play *Game of Life* on the Robinson Triangle tiling, we create several functions using a programming language called `Julia`. [4]

1. Substitution method

The Robinson Triangle tiling can be created using a *substitution* or *inflation and subdivision* procedure. [5] We inflate each prototile by a constant factor and subdivide each into copies of prototiles. For the Robinson Triangle tiling, the inflation factor is $\varphi = \frac{1+\sqrt{5}}{2}$ and we subdivide each tile following the rules shown in Figure 2.

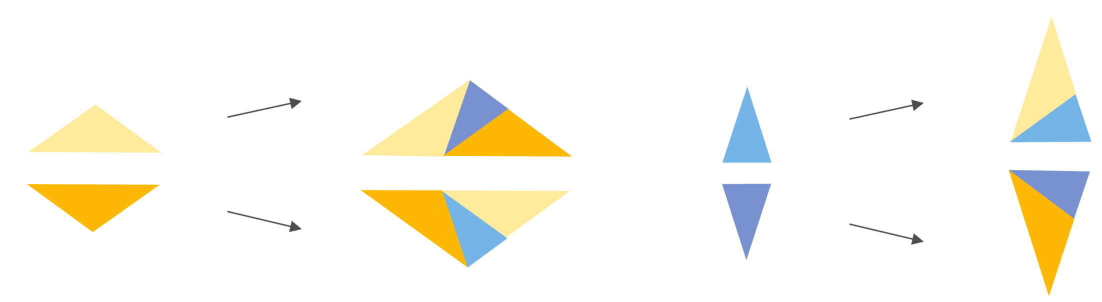

Figure 2: Substitution rule for prototiles in Robinson Triangle tiling.

2. Game of Life on Robinson Triangle Tiling

Step 1. Get Robinson Triangle tiling (`substitution_robinson_triangle_list`)

- Parameters: initial configuration `[tile type, position, orientation]`, substitution level, and a side length of initial tiles.
- Apply the substitution and get a list of tiles' information after *n* substitution. Use the output to get a graphic of the tiling (Figure 3).

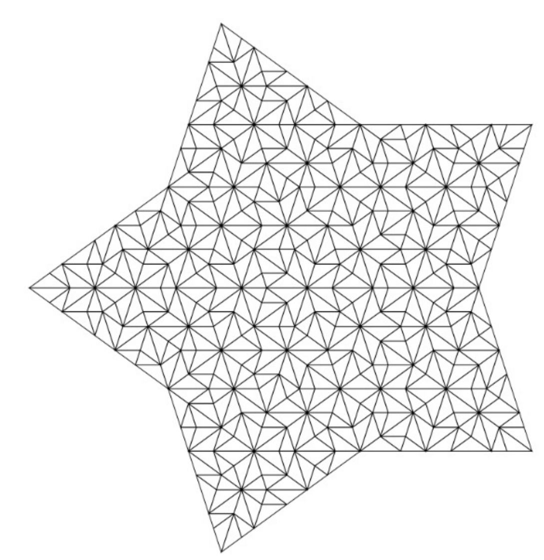

Figure 3: Level-4 Robinson Triangle tiling.

Step 2. Gather neighbor information (`neighbor_robinson_triangle_more_info`)

- Parameters: output of substitution function and coordinate list of tiles.
- Compare coordinates of each tile. If tiles share one coordinate, two tiles share one corner. If tiles share two coordinates, two tiles share an edge. Thus, they are neighbors, and we store those neighboring tiles' information in the output vector.

Step 3. Play *Game of Life* (`one_move, GoL_Robinson_triangle`)

- `one_move`: produce a list of live cells after one generation.
- Parameters: list of live cells for the current generation and neighbor information.
- Generate a state (0, 1) vector for current generation and get a sum of live neighbors. Apply *Game of Life* rules to generate list of live cells after one generation.
- `GoL_Robinson_triangle`: get lists of live cells after all generations.
- Parameters: initial live cells, neighbor information, and a maximum number of generations.
- Compare live cell list of current and next generation using `one_move`. Keep compare two lists until we reach a steady state or a maximum number of generations.
- Using the output, we generate an animation that shows the Game of Life procedure.

Results

As a result of playing Game of Life on the Robinson Triangle Tiling, we discover several still life patterns and oscillators by running functions we have. As a result, we find eight still life patterns (Figure 4) and two oscillators (Figure 5 and Figure 6) by playing Game of Life on level-6 substitution tiling.

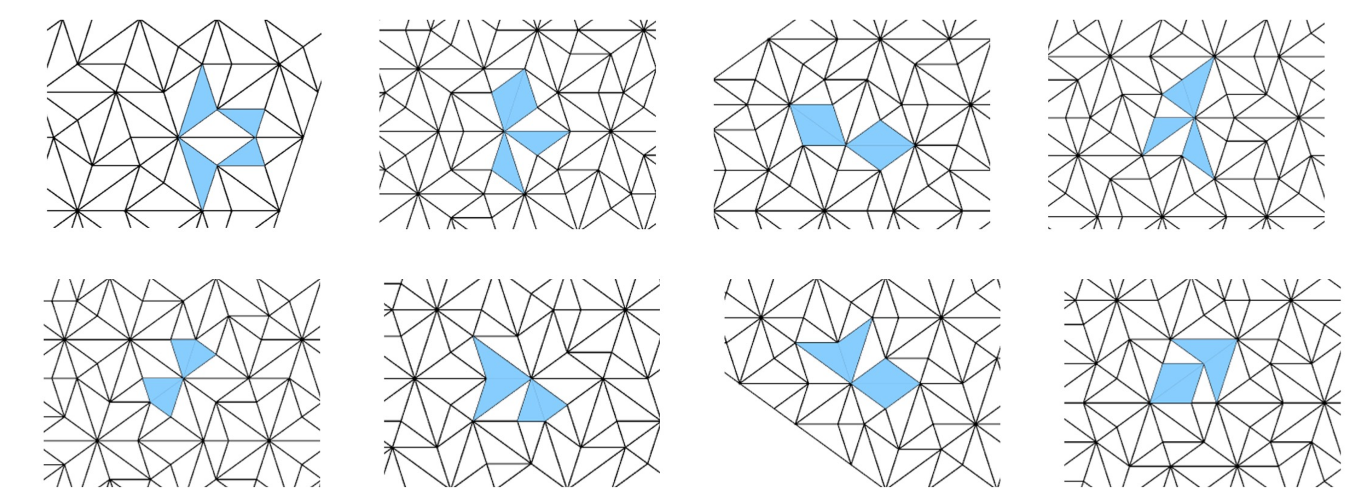

Figure 4: Eight still life patterns.

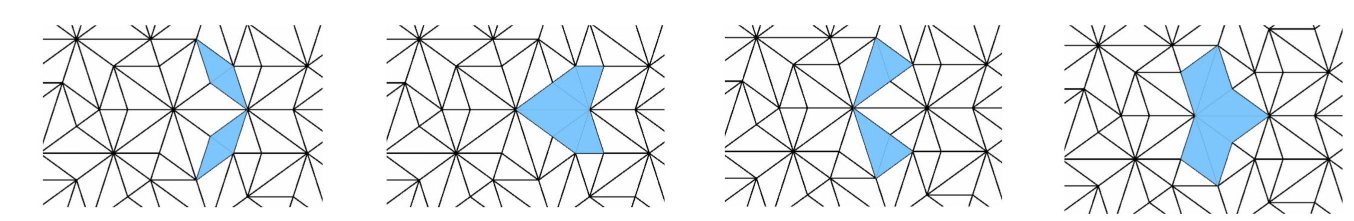

Figure 5: Period-4 oscillator.

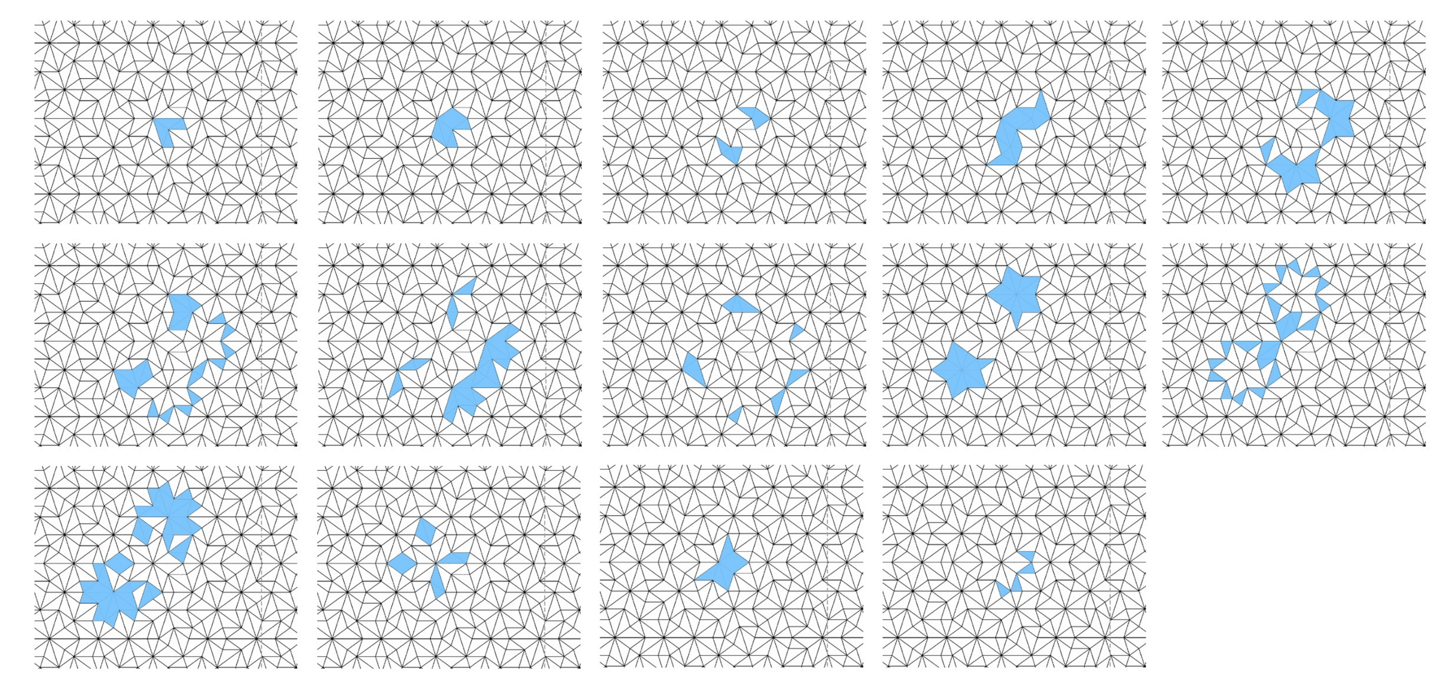

Figure 6: Period-14 oscillator.

Acknowledgements

J. Reid and Polly Anderson endowment funded this research, and we thank them for the support of this research. Also, we would like to thank Professor May Mei for the supportive mentorship and guidance. Lastly, I want to acknowledge Zheng Fang and Raghav Goel, who I worked together this summer, for great feedback and suggestions.

References

[1] Siobhan Roberts. The lasting lessons of john conway's game of life. `https://www.nytimes.com/2020/12/28/science/math-conway-game-of-life. html?searchResultPosition=1`, accessed June 1, 2022.

[2] Martin Gardner. Mathematical games: The fantastic combinations of john conway's new solitaire game "life". *Scientific American*, 223(4):120–123, 1970.

[3] Martin Gardner. Mathematical games: Extraordinary nonperiodic tiling that enriches the theory of tiles. *Scientific American*, 236(1):110–121, 1977.

[4] Julia Computing company. The julia programming language. `https://julialang. org/`, accessed January 28, 2022.

[5] Bielefeld University. Substitution. `https://tilings.math.uni-bielefeld.de/ glossary/substitution/`, accessed June 18, 2022.