Analysis of Recurrence Relations and Tile Combinatorics in Loop Hero Game

Angelina Andra Alanna - 13524079[1]

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

[1]angelinaandra93@gmail.com, 13524079@std.stei.itb.ac.id

***Abstract*— Loop Hero is a deck-building and strategy game where players place different tiles on a looping path to control the map, spawn enemies, and get resources. Many tile effects in the game follow patterns that can be explained using recurrence relations. Also, the number of ways place tiles can be analyzed using combinatorics. In this paper, we model some of these game mechanics using simple recurrence relations and count possible tile placements with combinatorics. This shows how ideas from discrete mathematics can help understand and plan better strategies in Loop Hero.**

***Keywords*— Combinatorics, discrete mathematics, Loop Hero, recurrence relations, tile placement.**

I. Introduction

Video games are a popular form of interactive entertainment that combine storytelling, art, and logic [1]. Many games use mathematical structure to create unique experiences. Other than for entertainment, video games can be a useful way to study computational and mathematical problem[1]. Some game mechanics are dependent on mathematics, either on purpose or by accident. By using discrete math on the game mechanism, we can better strategize ways to solve in game problems [1].

Loop Hero is a rogue-lite deck-building game developed by Fout Quarters and published by Developer Digital [2]. The game centers on the concept of an endless loop, where players place tiles representing terrain, buildings, and enemies along a looping path to affect the behavior of the hero and the world. These tiles interact with each other and evolve based on the order and frequency of placement, introducing a complex web of mechanics that involve decision-making, probability, and pattern analysis [6].

One of the interesting aspects of *Loop Hero* is how the tile placements create repeating loops, similar to recurrence relations in math. Some combinations of tile lead to predictable cycles of resources, enemies, or tile changes. The different ways tiles can be arranged also resemble combinatorics problems, where finding the best or most efficient setup matters.

This paper will use recurrence relations and combinatorics to model the *Loop* Hero's tile mechanics. The goal is to represent in game events – like resource generation or enemy spawns – using mathematical sequences and to study tile arrangements using counting methods. By doing this, we can show how discrete math helps explain the strategies behind *Loop Hero*.

II. Theoretical Framework

A. Recurrence Relations

A recurrence relation is a way to define a sequence (a list of numbers) where each number depends on the ones before it [3]. For example, a simple one is:

$$a_n = a_{n-1} + d \tag{1}$$

This means that to get the value of a_n we add something (called d) do the previous value. In Loop Hero, many things happened repeatedly as time goes on, like enemies appearing every couple loops. Things like these can be explained using recurrence relations, because current events were determined by our previous steps.

Types of recurrence relation:

- Linear recurrence: A sequence that changes by adding or multiplying a fixed value.
- Homogeneous recurrence: Uses only the previous values, like $a_n = 2a_{n-1} + 3a_{n-2}$
- Non-homogeneous recurrence: Includes outside values, like $a_n = a_{n-1} + 2n$.

(2)

These are useful for showing how game mechanics repeat or grow over time.

B. Solving Recurrence Relations

Solving a recurrence means finding a formula to get the value of a term directly, without needing to calculate all the ones before it.

Sone ways to solve them:

- Iteration: Listing out a few terms to see a pattern.
- Substitution: Making a guess of the general form of the solution based on a pattern, then checking if that form fits the recurrence.
- Using characteristics equations: For more complex forms, usually in algorithms.
- Generating function: A more advanced way that uses series.

In this paper, we mostly use simple methods like iteration and substitution to explain *Loop Hero* mechanics.

C. Combinatorics

Combinatorics is a branch of mathematics that studies the

arrangement of objects. The solution obtained is the number of arrangements of certain objects in a set [3]. This is useful to know how many ways something can happen.
Some basic ideas:

- Permutation: How many times to arrange items in order.

$$P(n,r) = \frac{n!}{(n-r)!} \tag{3}$$

Example: Arranging 3 out of 5 different tiles

$$\rightarrow P(5,3) = \frac{5!}{(5-3)!} = \frac{120}{2} = 60\ ways$$

- Combinations: How many ways to choose items without caring about order.

$$C(n,r) = \binom{n}{r} = \frac{n!}{r!(n-r)!} \tag{4}$$

Example: Many ways to choose 2 out of 4 tiles

$$C(4,2) = \binom{4}{2} = \frac{4!}{2!(4-2)!} = \frac{24}{2\times 2} = 6\ ways$$

- Inclusion-Exclusion: A way to count when choices overlap.

In *Loop Hero*, combinatorics helps us find how many ways we can place tiles or trigger effects by putting tiles near each other.

D. Loop Hero Game Mechanics

Loop Hero is a game in which players place different tiles (like Rock, Meadow, Village) on a circle shaped path (loop), These tiles effects how the game progresses in such ways:

- Spawning enemies
- Giving resources
- Changing into stronger tiles if placed in certain ways (like 3x3 Mountains turning into a Peak) [4].

These effects aren't random they follow specific rules, and often repeat or build up over time. That's why we can use recurrence and combinatorics to study and model them.

Figure 1. Loop Hero
Source: gamesvillage.it

III. Problem Modeling

This chapter explain how some parts of *Loop Hero* can be modeled using recurrence relation and combinatorics. The goal is to show how to turn tile mechanics in the game into simple math problems that helps us understand them better.

A. Using Recurrence Relations to Model Tile Effects

In *Loop Hero*, some events happen in cycles or after a certain number of steps. These patterns can be described using recurrence relations, because the current result depends on how many times something happened before.

1. Mountain Peak Formation

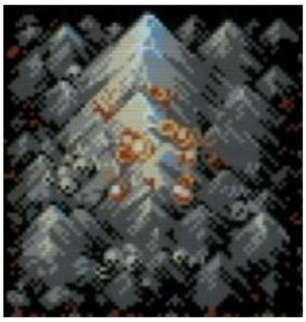


Figure 2. Mountain Peak Tile
Source: loophero.fandom.com

If you place a 3×3 group of Mountain or Rock tiles, it turns into a Mountain Peak. This happens every 9 tiles [4]. If we call the total number of tiles placed n, then the number of Mountain Peaks formed is:

$$M(n) = \left\lfloor \frac{n}{9} \right\rfloor \tag{5}$$

This means for every 9 tiles, you get 1 Mountain Peak. So, if you place 18 tiles, you get 2 Mountain Peaks [6].

2. Goblin Camp Spawning

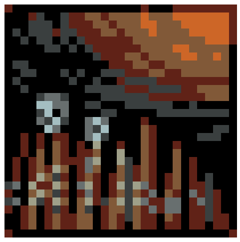


Figure 3. Goblin Camp Tile
Source: loophero.fandom.com

For every 10 Mountain or Rock tiles, a Goblin Camp appears and spawn enemies [4]. We can write it like this:

$$G(n) = \left\lfloor \frac{n}{10} \right\rfloor \tag{6}$$

This helps players know when to expect a new Goblin Camp.

3. Ransacked Village Transformation

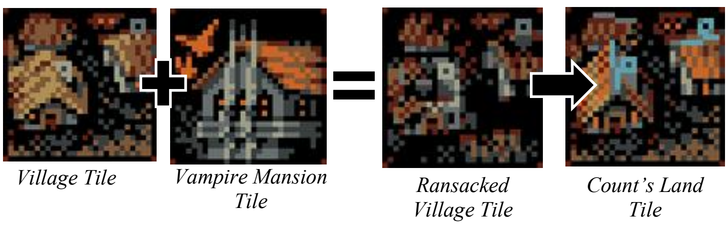


Figure 4. Ransacked Village Transformation
Source: loophero.fandom.com

If you place a Vampire Mansion next to a Village, it becomes a Ransacked Village. After 3 loops (rounds), it becomes a stronger tile called Count's Land [4][5]. We can write the transformation as:

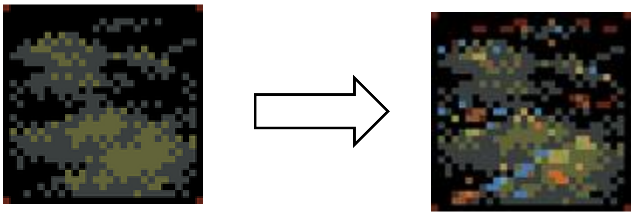
$$R(n) = \begin{cases} Ransacked\ Village, & if\ n < 3 \\ Count's\ Land, & if\ n \geq 3 \end{cases} \tag{7}$$

This helps us model how tiles change over time depending on how many loops have passed.

B. Using Combinatorics to Analyze Tile Placements

Tile placement is an important part of *Loop Hero*. Where you place each tile can give better rewards or stronger enemies. We can use combinatorics to count how many ways we can place tiles and what effects happen.

1. Blooming Meadow



Meadow Tile *Blooming Meadow Tile*

Figure 5. Blooming Meadow Transformation
Source: loophero.fandom.com

Meadow tiles normally restore a small amount of HP at the start of each day. However, if a Meadow is placed adjacent to any tile that is not another Meadow it becomes a Blooming Meadow, which heals the hero HP by 3 at the start of each day (loop) [2][4]. This transformation depends on tile adjacency, making it a spatial rule that can be analyzed using combinatorics and neighborhood logic in a 2D grid.

Let's model this transformation using a simple rule: If a Meadow tile is placed at position (i, j), and it has at least one neighboring tile (orthogonally or diagonally) that is not a Meadow, then it becomes a Blooming Meadow. On a square grid, each tile has 8 possible neighbors (up, down, left, right, and the 4 diagonals). We can represent the number of non-Meadow tiles surrounding a Meadow as b, where $0 \leq b \leq 8$.
Let's define a binary condition:

$$Blooming(i, j) = \begin{cases} 1, if\ at\ least\ one\ adjacent\ tile\ is\ not\ a\ Meadow \\ 0, otherwise\ (surrounded\ only\ by\ Meadows\ or\ empty) \end{cases} \tag{8}$$

So, tile becomes a Blooming Meadow if and only if:

$$b \geq 1$$

Assume a player want to place m Meadow tiles. The number of Blooming Meadows that can be created depends on how many of those are adjacent to at least one non-Meadow tile. Let's say the player has t total other tiles (non-Meadows) on the board. Each of those t tiles creates up to 8 adjacent spots where a Meadow could be placed next to it, thus triggering the Blooming effect.

So, the maximum number of Blooming Meadows can be estimated as:

$$\min(m, 8 \times t) \tag{9}$$

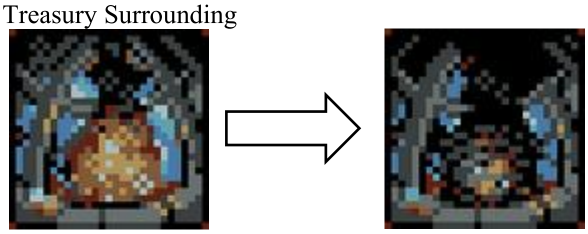
Where:

- m is the number of Meadow tiles
- t is the number of adjacent non-Meadow tiles already placed

This is a simplified estimate that assumes no tile overlap and perfect placement.

This mechanic creates an interesting combinatorics problem: how to place Meadow tiles in a way that maximizes the number of Blooming Meadows. Players must place Meadows adjacent to other tile types — but not too close to other Meadows — which involves planning space usage carefully, similar to solving a tiling puzzle on a constrained grid [5].

2. Treasury Surrounding



Treasury Tile *Empty Treasury Tile*

Figure 6. Treasury Tile Transformation
Source: loophero.fandom.com

The Treasury tile gives one reward each time a player places a tile in one of the 8 spaces surrounding it. This continues until all 8 adjacent spaces are filled. After that the treasury tile would transform into an empty treasury tile, which no longer gives any rewards [4]. Let f be the number of filled surrounding tiles. Then the total rewards form a Treasury can be modeled as:

$$R(f) = \begin{cases} f, if\ 0 \leq f < 8 \\ 8, if \geq 8\ (beomes\ Empty\ Treasury) \end{cases}$$

This mechanic encourages players to place tiles gradually and plan their map carefully to maximize resources.

To understand how many possible ways a player can surround a Treasury with tiles, we apply a combinatorics approachs.

There are approximately 45 different tile types that the player can place during a run [2]. Each of the 8 surrounding positions can be filled with any of these tile types. In addition:

- The same tile can be used more than once (e.g., placing multiple Meadows)
- The order matters, because the location of each tile affects its interaction

Because repetition is allowed and position matters, this problem uses the formula for permutations with repetition:

$$n^r$$

Where:

$n = 45$ $(number\ of\ tile\ types)$
$r = 8$ $(number\ of\ surrounding\ positions)$

Which means the total number of possible surrounding tile combination is:

$$45^8 = 1.68 \times 10^{13}$$

This is over 16 trillion different arrangements, which is far too many to analyze directly.
To simplify the problem, we can group the tile types into categories based on function such as [2]:

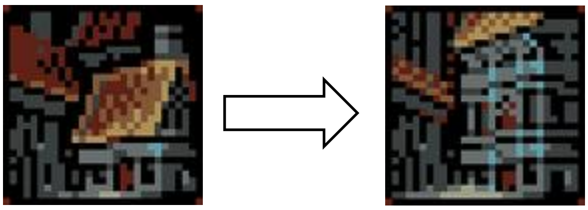
- Terrain tiles (e.g., Rock, Meadow)
- Enemy-spawning tiles (e.g., Swap, Vampire Mansion)
- Buff-giving tiles (e.g., Beacon, Blood Grove)
- Support tiles (e.g., Village, Suburb)
- Utility/Neutral tiles (e.g., Bookery, Outposts)

Using these 5 categories, the number of possible combinations becomes:

$$5^8 = 390{,}625$$

This is still a large number, but much more manageable for analysis. It also reflects the strategic choices players makes when surrounding a Treasury, not just how many tiles are placed but what kind and where.
This example shows how combinatorics can be used to understand game mechanics and help players make better decision based on tile placement patterns.

3. Suburb Upgrade



Suburbs Tile *Town Tile*

Figure 7. Suburb Upgrade Transformation
Source: loophero.fandom.com

In Loop Hero, Suburb tiles increase the hero's experience gain rate. However, if Suburbs are arranged in a specific shape, a plus (+) formation, where one Suburb is surrounded on four sides (top, bottom, left, right) by other Suburbs, the central tile is transformed into a Town, which gives even more experience [4].

This Transformation introduces a spatial pattern that can be analyzed using combinatorics and grid logic. Each Town requires exactly 5 Suburb tiles, with one at the central and four directly adjacent. Importantly, diagonal placement does not count.

Mathematically, if we represent the map as a 2D grid and consider a given tile at position (i, j), then a Town forms if:

$$Suburb(i,j)\ and\ Suburb(i-1,j)\ and\ Suburb\ (i+1,j)\ and\ Suburb\ (i,j-1)\ and\ Suburb\ (i,j+1) \quad (10)$$

This condition represents a local neighborhood constraint and can be interpreted as a small pattern-matching problem on a grid.

From a combinatorics perspective, if a player has s Suburb tiles available and wants to calculate maximum numbers of Towns that can be formed, they must consider:

- Each Town consumes 5 Suburb tiles
- A Suburb tile can only be used in one Town center, because the center is unique

So, in the best case (assuming perfect placement without overlap), the upper bound on Towns is:

$$T(s) = \left\lfloor \frac{s}{5} \right\rfloor \quad (11)$$

Each Town require 5 unique Suburb tiles arranged in a plus (+) shape. Suburbs cannot be reused across multiple Towns [2], this makes Suburb placement a type of combinatorial optimization, where players must position tiles on a grid to maximize the number of valid Towns without wasting tiles. The challenge is similar to tiling in discrete math, where space and structure constraints must be balanced carefully.

C. Modeling Resource Gain

You also gain resources in the game by placing tiles or going through loops. This can also be modeled with simple math.

1. From Battles

Let's say you get r resources per loop. Then after n loops:

$$R(n) = r \times n \quad (12)$$

2. From Tile Placement

If each tile placed give some resource (like Rock giving Pebbles), then placing n tiles gives:

$$P(n) = p \times n \quad (13)$$

These two models can be combined to estimate the total resources a player gains during a run. This helps in planning strategies for upgrades and progression, such as determining how many loops or tiles are needed to unlock buildings in the camp.

IV. Analysis and Discussion

This chapter discusses how the mathematical models in Chapter III can help explain tile mechanics in the game Loop Hero. It focuses on how these models support strategic decision-making during gameplay, and how similar mathematical ideas also appear in real-world systems.

A. Recurrence Relations and Game Timing

The recurrence relations presented earlier allow players to predict when game events will occur. For example, the appearance of Goblin Camps or the transformation of a Ransacked Village can be modeled as functions that depend on counts or loops. These patterns give the player control over when and where tile-based effects will happen.

Such models are useful because:

- They allow strategic placement of tiles to trigger effects at the right time.
- They help players avoid unintended consequences, such as spawning too many enemies too early.

- They mirror real-world timed systems, such as repeated events in software(loops) or maintenance cycles in machines.

By using recurrence relations, players can think ahead and optimize outcomes over time, just like planning in programming or logistics.

B. Combinatorics and Tile Arrangement Strategy

Combinatorics helps describe how many ways tile can be placed, and which patterns create stronger or more useful effects. In *Loop Hero*, this is especially important for tile combinations like:

- Surrounding Treasury to gain loot
- Arranging Suburb into Towns to gain bonus experience.
- Creating Blooming Meadows through adjacency

The calculations in Chapter III shows just how many possible placements there are. However, not all of them are efficient or valid, which makes planning just that much more important. Combinatorics helps player:

- Count possibilities
- Eliminate useless arrangements
- Find the best placement for maximum benefits

This kind of thinking is also useful outside of games – such as arranging seats, planning networks, or solving design puzzles.

C. Modeling Resource Gain and Efficiency

The arithmetic models for resource gain gives a simple way to estimate the progress a player will make in the run. Whether from placing tiles or surviving loops, the formulas help answer questions like:

- How many loops it takes to gather a certain material
- How many tiles placement it takes to achieve a target

Therefore, players can manage resources more efficiently. Rather than just guessing, math can be a way to predict outcomes and plan better.

In real life, this is similar to:

- Counting income over time
- Planning supply in factories or stores
- Managing budgets or production rates [5]

So even though the models are simple, they give useful insight into both game and practical planning systems.

D. Connections to Real-World Systems

The mathematical concepts used in *Loop Hero* are not only relevant in games – they also exist in many areas of daily life.

1. Recurrence Relations in Real Life
   - A store restocking every 3 days, or
   - A machine that needs maintenance every 100 hours [5]

These are examples of step-based actions, just like enemy spawning or tile transformation in the game. Using recurrence relations helps in planning and scheduling these actions efficiently

2. Combinatorics in Real Life
   - Choosing members for a team
   - Arranging furniture or parts on a grid
   - Designing circuits or layouts [5]

All these problems involve counting possibilities and choosing the best arrangements. Whether in game or real life, combinatoric helps reduce trial and error and improve outcomes.

E. Summary of Insights

From the discussion above, we can summarize that:

- Recurrence relation helps model when an event occurs.
- Combinatorics helps analyze how tile setups affect gameplay
- Arithmetic models help players to predict resource gain and manage progress.
- All these ideas are not only useful in games – they also reflect systems used in business, engineering, and daily planning.

This shows that discrete mathematics is a powerful tool, even in unexpected places like game design. It helps players become more strategic, and gives a deeper appreciation of the systems behind the game.

V. Conclusion

This paper has demonstrated how recurrence relations and combinatorics can be used to model and analyze tile mechanics in the game Loop Hero. Recurrence relations allow us to predict when certain in-game events occur, such as enemy spawns or tile transformations, based on placement patterns or loop counts. Meanwhile, combinatorics helps analyze the number of possible tile arrangements and supports strategic planning to optimize outcomes.

These mathematical models not only help players make better decisions in the game, but also reflect real-world systems in fields such as scheduling, inventory control, and spatial layout optimization. This highlights that discrete mathematics is not only a theoretical subject, but also a practical tool for analyzing and solving real problems — including those found in games. Future research could explore more advanced tile interactions, model conditional tile synergies using logical systems, or develop algorithms to detect optimal placements based on the underlying mathematics.

V. Acknowledgment

First and foremost, the author expresses gratitude to Allah SWT for the strength and guidance given throughout the writing of this paper. The author would also like to sincerely thank Dr. Ir. Rinaldi Munir, M.T., as the lecturer of IF1220 Discrete Mathematics, for his guidance and dedication during the course. The author also wishes to thank their family for the continuous support and motivation throughout the learning and writing process, as well as classmates for the helpful discussions and encouragement.

References

[1] S. Doble, "The Importance of Video Games," Project Brainlight, Sep. 12, 2021. [Online]. Available: https://www.projectbrainlight.org/blog/learn-by-gaming. [Accessed: Jun. 17, 2025].

[2] Community contributors, "Tiles," Loop Hero Wiki. [Online]. Available: https://loophero.fandom.com/wiki/Tiles. [Accessed: Jun. 17-19, 2025].

[3] R. Munir, *Matematika Diskrit (IF1220) - Teori Relasi Rekurensi dan Kombinatorika*, Institut Teknologi Bandung, 2024. [Online]. [Accessed: June 17-18, 2025].

[4] S. Jain, "Loop Hero – All Tile Combos," *TheGamer*, May 2021. [Online]. Available: https://www.thegamer.com/loop-hero-all-tile-combos-guide/. [Accessed: June 17-19, 2025].

[5] K. H. Rosen, *Discrete Mathematics and Its Applications*, 7th ed., New York: McGraw-Hill, 2012.

[6] Loop Hero Developers (Four Quarters), "Loop Hero – PC Game," *Devolver Digital*, 2021. [Online]. Available: https://store.steampowered.com/app/1282730/Loop_Hero/. [Accessed: Jun. 17-19, 2025].

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 19 June 2025

Angelina Andra Alanna 13524079