DESIGNING A TWO-PLAYER MAJORITY COLORING GAME WITH A USER INTERFACE

by

Gülce Baysal

B.S., Electronics and Communication Engineering, Istanbul Technical University, 2022

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Industrial Engineering
Boğaziçi University
2025

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my esteemed advisor, Prof. Tınaz Ekim, who has consistently guided me throughout this thesis with her profound knowledge, thoughtful approach, and patient support, and who has always been a source of inspiration for me. Her openness to sharing, her willingness to guide at every stage, her constructive and realistic feedback, and her genuine interest made this process both productive and enjoyable. I consider it a great privilege to have had the opportunity to work under her supervision, and I remain truly grateful for her invaluable contributions.

My deepest gratitude goes to my mother, Gülcan Ofluoğlu Baysal, for her unconditional love, constant support, and inspiring presence. I am also truly thankful to my father, Cafer Baysal, whose belief in me, strength, and integrity have made him my greatest role model. I would like to thank my sister, Damla Baysal Bakır, and her husband, Eren Ozan Bakır, for their encouragement, support, and unwavering faith in me. Special thanks to my niece, Mavi Derin Bakır, whose joyful presence and the desire to be a good role model for her have continuously motivated me throughout my academic journey.

Finally, I am profoundly grateful to Mustafa Kemal Atatürk, whose unwavering belief in the power of reason, science, and education paved the way for countless generations to dream, question, and freely pursue knowledge. His vision continues to remind me that every step taken in the light of learning lays the foundation for a better future.

ABSTRACT

DESIGNING A TWO-PLAYER MAJORITY COLORING GAME WITH A USER INTERFACE

This thesis examines a Majority Coloring Game in which two players alternately color the vertices of a graph. In this game, one player (Alice) attempts to color the graph in accordance with the majority rule, while the other player (Bob) tries to sabotage this process. The study develops algorithmic strategies such as Greedy, Stackelberg, and OddReduction, and provides a detailed analysis of how structural properties, color constraints, the number and placement of odd-degree vertices, and the competitive interactions between players affect the level of game difficulty. Various graph families, including cycle, wheel, Petersen, and cubical graphs, were experimentally investigated. The effects of graph size, vertex degree distribution, and the number of available colors on game complexity were also evaluated. In addition, an interactive game interface was developed using the Unity game engine, enabling real-time gameplay against an AI opponent (Bob) employing different strategies. This platform serves both as a pedagogical tool for teaching graph theory and the majority coloring approach and as an experimental environment for validating theoretical findings. The results demonstrate that game complexity is significantly influenced by graph topology, the arrangement of odd-degree vertices, color constraints, and competitive dynamics. These findings provide valuable insights into the relationship between strategic decision-making and structural factors.

ÖZET

İKİ OYUNCULU ÇOĞUNLUK BOYAMA OYUNUNUN KULLANICI ARAYÜZÜ İLE TASARIMI

Bu tez, iki oyuncunun sırayla bir grafın düğümlerini boyadığı bir çoğunluk boyama (majority coloring) oyununu incelemektedir. Bu oyunda bir oyuncu (Alice), grafı çoğunluk kuralına uygun şekilde boyamaya çalışırken diğer oyuncu (Bob) bu süreci sabote etmeye çalışır. Çalışmada Greedy, Stackelberg ve OddReduction gibi algoritmik stratejiler geliştirilmiş; yapısal özellikler, renk kısıtları, tek dereceli düğümlerin sayısı ve yerleşimi, oyuncular arasındaki rekabetçi etkileşimler gibi unsurların oyunun zorluk düzeyi üzerindeki etkileri ayrıntılı şekilde analiz edilmiştir. Çevrim, tekerlek, Petersen ve kübik graf gibi farklı grafik aileleri deneysel olarak incelenmiş; graf boyutunun, düğüm derece dağılımının ve kullanılabilir renk sayısının oyun karmaşıklığına etkileri değerlendirilmiştir. Ayrıca Unity oyun motoru kullanılarak gerçek zamanlı olarak farklı stratejilerle hareket eden yapay zekâ rakibe (Bob) etkileşim sağlanabilen interaktif bir oyun arayüzü geliştirilmiştir. Bu platform, hem grafik kuramı ve çoğunluk boyama yaklaşımını öğretmeye yönelik pedagojik bir araç hem de teorik bulguların doğrulanmasına imkân tanıyan bir deneysel ortam işlevi görmektedir. Elde edilen sonuçlar, oyunun karmaşıklığının graf topolojisi, tek dereceli düğümlerin düzeni, renk kısıtları ve rekabetçi dinamiklerden önemli ölçüde etkilendiğini göstermekte; stratejik karar alma süreçleri ile yapısal faktörler arasındaki ilişkiye dair değerli içgörüler sunmaktadır.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

C	Set of colors available
C_n	Directed cycle graph
D	Digraph (directed graph)
E	Set of edges in the graph
$G = (V, E)$	Graph with vertex set V and edge set E
P_m, P_n	Directed path graphs
S_n	Star graph with n vertices
V	Set of vertices in the graph
$c(v)$	Color assigned to vertex v
$col(G)$	Coloring number
$col_g(G)$	Majority game coloring number
$d(v)$	Degree of vertex v
k	Integer parameter representing the maximum back degree in a vertex ordering
α	Parameter indicating almost-regularity of a digraph in majority coloring conditions
$\chi(G)$	Chromatic number of a graph G
$\chi'(G)$	Game chromatic index: minimum number of colors needed for an edge-coloring game on graph G
δ^+	Minimum outdegree in a digraph
$\Delta(G)$	Maximum degree of the graph
ε	Small positive constant used in conjectures (e.g., asymptotic bounds)
$\mu(G)$	Majority chromatic number
$\mu_g(D)$	Majority game chromatic number for digraph D
$\mu_g(G)$	Majority game chromatic number

LIST OF ACRONYMS/ABBREVIATIONS

AI	Artificial Intelligence
AVG	Average
OD	Odd Degree
UB	Upper Bound
UI	User Interface
URL	Uniform Resource Locator
WebGL	Web Graphics Library

1. INTRODUCTION

This thesis examines the Majority Coloring Game by analyzing its structural dynamics, defining graph-theoretic difficulty levels, and developing an interactive two-player framework. Formally, a coloring is called a majority coloring if no vertex has more than half of its neighbors in the same color as the vertex itself. This concept has practical applications in areas such as network load balancing, modeling social interactions, and designing fair voting systems, where balanced influence or resource distribution is essential.

Building on the existing theoretical foundations of majority coloring, this study develops original strategy models to better reflect its dynamic and adversarial nature. By incorporating a game-theoretic perspective, it explores strategic coloring decisions on various graph types with different structural properties and demonstrates how graph size, the proportion and placement of odd-degree vertices, and color constraints fundamentally shape the level of difficulty.

In addition to the theoretical framework, this thesis systematically tests how three original strategies, namely Greedy, Odd-Reduction and Stackelberg, which were developed specifically within the scope of this research, affect Alice's ability to construct a valid majority coloring in response to Bob's attempts to create rule violations. The experimental design includes both classical and specially constructed graph types that vary in size, degree distribution and the placement of structurally critical nodes, providing a comprehensive understanding of the dynamics of the majority coloring problem.

Finally, an interactive interface was developed to simulate gameplay, visualize majority rule constraints and support real-time experimentation. This platform allows parameter adjustments and provides dynamic visual feedback. As a result, it serves not only as a flexible research tool but also as an educational resource for teaching graph theory and strategic thinking.

1.1. Problem Definition

A game theoretic variant of majority coloring, defined via a two-person game, is as follows. The board of the game is a fixed graph G. Two players, Alice and Bob, are coloring vertices of G alternately, with Alice playing first, using either a fixed set of colors C or with a potentially unbounded set of colors that is dynamically expanded during gameplay. Throughout the game, the partial coloring must comply with the majority rule after every move. This obliges Alice to maintain the rule, whereas Bob's main objective is to create a violation. Alice wins if the whole graph is eventually colored; Bob wins when a partial coloring arises that cannot be further extended without violating the majority condition.

There are certain parameters that are related to coloring games. The chromatic number $\chi(G)$ is the smallest integer k for which there exists a k-coloring of G, meaning a coloring of the vertices using k colors such that adjacent vertices always have different colors. Additionally, the coloring number of a graph G, denoted as $\mathrm{col}(G)$, is closely related to the concept of back degree. Specifically, $\mathrm{col}(G)$ is defined as

$$\mathrm{col}(G) = k + 1, \tag{1.1}$$

where k is the smallest possible maximum back degree that can be achieved across all possible linear orderings of the vertex set V. In other words, there exists an optimal linear ordering in which each vertex has at most k neighbors that appear earlier in the order. To determine $\mathrm{col}(G)$, we find such an ordering that minimizes the maximum back degree. By finding the smallest achievable maximum back degree over all possible orderings, the coloring number reflects how well the graph can be arranged to minimize backward connections.

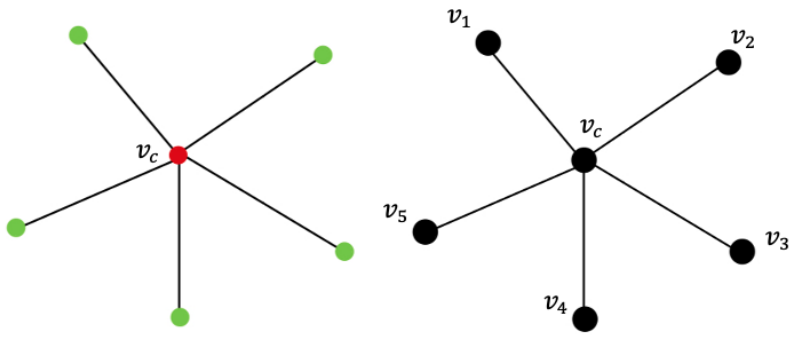


Figure 1.1. Chromatic number and coloring number.

A star graph S_n consists of a central vertex connected to n leaf vertices, such as S_5, which has one central vertex and five leaves. The chromatic number of a star graph, $\chi(S_5)$, is 2, as only two colors are required as seen in the left of Figure 1.1: one for the central vertex and another shared by all leaf vertices since they are not adjacent to each other. In contrast, in tree-like structures such as a star graph, it is crucial to remember that the coloring number is defined based on the minimum possible maximum back degree across all linear orderings of the vertices. For example, in a star graph (e.g., S_5), placing the central vertex at the beginning of the ordering minimizes back degrees: the central vertex then has a back degree of zero, while each leaf vertex has a back degree of one, yielding $\mathrm{col}(G) = 1 + 1 = 2$. However, if the central vertex is placed at the end of the ordering, it will have a back degree equal to the number of leaf vertices (in this case, five), resulting in a maximum back degree of five and a local calculation of $\mathrm{col}(G) = 1 + 5 = 6$ for that specific ordering. Despite this, the coloring number always uses the optimal ordering with the smallest achievable maximum back degree. Therefore, for any star graph, the coloring number remains 2, as the best possible arrangement ensures that the maximum back degree does not exceed one. While the chromatic number reflects the minimum colors needed for proper coloring, the coloring number captures the constraints imposed by a specific vertex ordering, highlighting their difference and the impact on strategies in coloring games.

The parameter $\mu(G)$ denotes the theoretical majority chromatic number of a given graph G, which represents the minimum number of colors required to color the graph while ensuring that no vertex has more than half of its neighbors in the same color as the vertex itself. In the context of the Majority Coloring Game specifically, the game variants of these parameters are denoted as $\mu_g(G)$ and $\text{col}_g(G)$, reflecting the fact that these values are influenced by the adversarial interaction between the two players. The majority game chromatic number $\mu_g(G)$ of a graph $G = (V, E)$ is the least cardinality of a color set X for which Alice has a winning strategy. As seen in Figure 1.2, in a scenario where Alice colors v_1 red in her first move and Bob subsequently colors v_2 green, Alice is forced to color w_1 with a third distinct color in her second move. Otherwise, Bob, in his second move, would color x the same as w_1, thereby breaking the majority coloring rule and defeating Alice. In the graph shown in Figure 1.2, $\mu_g(T) \leq 3$.

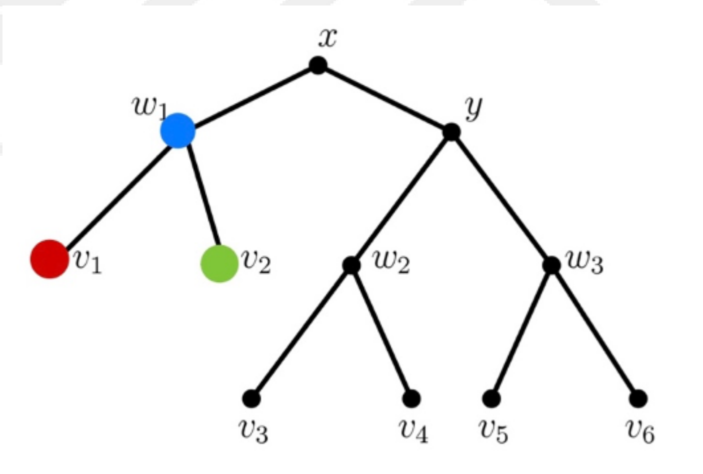


Figure 1.2. Majority coloring game example.

Meanwhile, the majority game coloring number, denoted as $\text{col}_g(G)$, is defined as the smallest integer $k+1$ such that Alice has a strategy to keep the number of backward neighbors of each vertex strictly below $k+1$. Equivalently, this means that the maximum back degree is at most k, resulting in $\text{col}_g(G) = k+1$. In the Majority Coloring Game, the concept of back degree becomes a dynamic parameter since vertex selection order is adversarial and directly affects the majority rule condition. Figure 1.3 illustrates an example of majority coloring. Assuming Alice starts the game, odd-numbered vertices in the linear order can be viewed as selected by Alice and evennumbered vertices by Bob. When Alice chooses to color the 7th vertex, it has four backward neighbors in the order (v_1, v_4, v_5, v_6). Therefore, for this specific linear order, the maximum back degree is 4, implying $\mathrm{col}_g(G) = 5$ for this scenario.

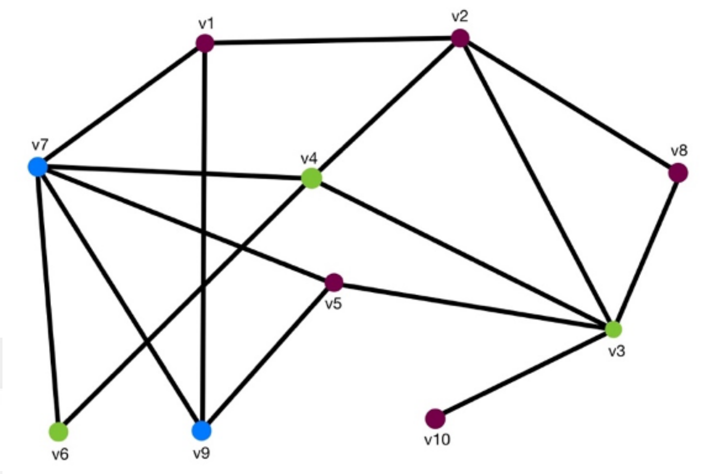


Figure 1.3. Majority coloring game scenario.

At first glance, the definition of the majority game coloring number may appear similar to the classical coloring number, since both are based on finding the minimum possible maximum back degree plus one. However, the key difference lies in the dynamic, adversarial nature of the Majority Coloring Game. In the classical setting, the linear ordering of vertices is fully optimized by the player to minimize the back degree, with no interference. In contrast, the Majority Coloring Game involves two players: Bob, who aims to maximize the maximum back degree to increase the risk of majority rule violations, and Alice, who seeks to minimize it to ensure the majority condition holds throughout the game. This adversarial interaction means that the back degree becomes a dynamic parameter directly linked not only to the vertex ordering but also to how colors are assigned under the majority rule constraint. While Alice tries to maintain a defensive strategy to keep the number of same-colored neighbors below the majority threshold, Bob intentionally selects and colors vertices to force risky configurations that could trigger a rule violation. Therefore, although the formal definition given in Equation (1.1) resembles the classical coloring number, the underlying strategic context is fundamentally different due to the majority rule and the adversarial moves shaping the achievable back degree.

2. LITERATURE REVIEW

The concept of majority coloring has been studied in various contexts, particularly in relation to minimizing monochromatic edges under coloring constraints. One of the foundational results in this area shows that every finite graph can be majority-colored using only two colors, as formalized

$$\mu(G) \leq 2. \tag{2.1}$$

This classical bound, given in Equation (2.1), was first established by Lovász [1].By employing a 2-coloring strategy that minimizes the number of monochromatic edges, we approximate an ideal distribution where each vertex has a majority of its neighbors in the opposite color. The theorem in Equation (2.1), which states that every finite graph G can be majority-colored using only two colors, can be understood through the concept of monochromatic edges. A monochromatic edge is an edge whose endpoints share the same color under a given vertex coloring [2]. In the context of majority coloring, the goal is to ensure that no vertex v has more than half of its neighbors in the same color, effectively limiting the number of monochromatic edges connected to v. This result guarantees that two colors are sufficient to achieve this balance for any finite graph, even in the worst-case scenario. The proof often involves analyzing the distribution of monochromatic edges in a random or iterative coloring process, showing that their number can be reduced until the majority condition is satisfied for all vertices. By minimizing monochromatic edges, this theorem highlights how edge-based constraints directly influence the feasibility of majority coloring, ensuring that the condition holds universally with only two colors. This method not only reduces monochromatic edges but also naturally achieves a majority coloring, ensuring that each vertex has most of its neighboring vertices in a color different from its own. Although this rule applies to all general graphs, it is not possible to satisfy this condition in the game variant version, where the game is played by two players, due to the numerous possibilities for moves.

Another important parameter for coloring games is the game coloring number. For any finite graph $G = (V, E)$, the game coloring number $\mathrm{col}_g(G)$ provides an upper bound for the game chromatic number $\mu_g(G)$, as every graph satisfies

$$\mu_g(G) \leq \mathrm{col}_g(G). \tag{2.2}$$

In the context of the game coloring number, two players, Alice and Bob, construct a linear order of the vertices by alternately selecting them. Alice's aim is to minimize the back degree, while Bob tries to maximize it. Once the order is completed, the game coloring number is defined as $1 + k$, where k is the maximum back degree in the game-created order. This arrangement also directly relates to the majority condition, as Alice ensures that for any uncolored vertex u, the number of colored neighbors is at most k. Since $\mathrm{col}_g(G)$ guarantees that $k+1$ colors suffice to manage all vertices without violating the majority condition, it follows that $\mu_g(G)$, the smallest number of colors needed to majority-color the graph, cannot exceed $\mathrm{col}_g(G)$. Thus, the relationship in Equation (2.2) is inherently satisfied [3].

However, it is important to note that in the Majority Coloring Game studied here, the vertex selection and the coloring process occur simultaneously, meaning that the distribution of colors must also comply with the majority rule condition at every stage of play. This additional constraint distinguishes the majority game coloring number from the classical marking game: while both rely on minimizing the maximum back degree, the majority game inherently integrates the dynamic risk of majority rule violations within the adversarial sequence of vertex selections and color assignments.

For planar graphs, $\mu_g(G) \leq 17$, and for complete binary trees, $\mu_g(T) \leq 3$, which is optimal for such trees. However, for bipartite graphs and graphs with $\mathrm{col}(G) = 3$, $\mu_g(G)$ is shown to be unbounded, revealing that even low-coloring-number graphs can have arbitrarily large majority game chromatic numbers [4].

These results underscore the complexity and unpredictability of the Majority Coloring Game under adversarial strategies. According to Anholcer, Bosek, and Grytczuk [5], every countable digraph D can be majority-colored using at most five colors ($\mu_g(D) \leq 5$) by partitioning the vertex set into subsets and applying tailored coloring strategies to satisfy the majority condition.

Wang and Wu [6] proved that every α-almost-regular digraph D with minimum outdegree

$$\delta^+ \geq \max\{740,\ 56 \ln(12\alpha)\} \tag{2.3}$$

has a majority 3-coloring. This result further refines the conditions under which majority 3-colorability is guaranteed, emphasizing the role of graph regularity.

Shi et al. [7] studied the majority coloring of the join and Cartesian product of specific digraphs. They proved that for the join of two directed paths P_m and P_n, the graph is majority 2-colorable; and for the join of a directed path P_m and a directed cycle C_n, the graph is majority 2-colorable if n is even and majority 3-colorable if n is odd.

In the general context of graph coloring, various bounds have been established for the game chromatic number and the game coloring number across different graph classes. Prior research has determined exact values of the game chromatic number for several well-known families, including paths, cycles, stars, wheels, complete graphs, complete bipartite graphs, and the Petersen graph. This line of investigation has also been extended to Cartesian products of graphs, such as combinations of paths and wheels, where existing bounds have been improved using the game coloring number, a closely related parameter. These findings highlight significant progress in identifying game chromatic numbers for specific graph structures. However, the literature emphasizes that many graph constructions, particularly Cartesian products of arbitrary graphs, remain open problems, presenting ongoing challenges and opportunities for further research [8].

Bartnicki and Grytczuk [9] explore an alternative approach to graph coloring games by focusing on edge coloring rather than vertex coloring. They introduce the game chromatic index $\chi'(G)$, representing the minimum number of colors Alice needs to win an edge-coloring game against Bob. Their research demonstrates that for graphs with arboricity k, the game chromatic index satisfies

$$\chi'(G) \leq \Delta(G) + 3k - 1, \tag{2.4}$$

where $\Delta(G)$ is the graph's maximum degree. For planar graphs ($k = 3$), this implies:

$$\chi'(G) \leq \Delta(G) + 8. \tag{2.5}$$

Their novel activation strategy avoids the need for ordering vertices or edges, instead managing the number of edges Alice can control during the game. The paper also raises open questions, such as finding tighter bounds for specific graph classes (e.g., complete graphs and trees) and understanding the asymptotic behavior of $\chi'(G)$ in dense graphs. While they conjecture that

$$\chi'(G) \leq (2 - \varepsilon)\Delta(G) \tag{2.6}$$

holds for all graphs with maximum degree $\Delta(G)$, the bounds may depend on graph density and structure. This work provides significant insights into edge-coloring games and lays a foundation for further studies on the game chromatic index across various graph families.

In conclusion, studies on coloring games have made significant contributions regarding various parameters such as the chromatic number, coloring number, game chromatic number, and game coloring number. While these relationships have been extensively documented, the Majority Coloring Game introduces a new layer of complexity by combining traditional coloring rules with the majority condition under competitive settings. Although upper bounds have been established for certain graph families, many structural and strategic aspects of the game remain largely unexplored.

Notably, there is no existing research that examines a dynamic, two-player version of the problem in which players take turns coloring vertices while preserving the majority constraint. This study fills that gap by developing a game-theoretic model for two players and evaluating its strategic structure through simulations.

In addition, there is limited knowledge on how to define the game's fundamental parameters, such as strategic options, difficulty levels, and balance mechanisms. By exploring how theoretical constructs translate into practical gameplay, this research highlights their role in the development of balanced strategies and contributes to the design of an effective user interface that reflects the game's strategic dynamics.

3. INSTANCE GENERATION

In order to thoroughly analyze the logic of the Majority Coloring Game and effectively design its structure, it is essential to investigate how various parameters influence the gameplay. This requires the formulation and systematic testing of key parameters such as the alternating moves of Alice and Bob, the structural characteristics of the graph, its size, and specific properties of individual vertices. Understanding the logic of coloring games and analyzing their underlying parameters is instrumental in the development of a user interface for the Majority Coloring Game. Moreover, a thorough examination of these parameters facilitates a deeper comprehension of the game mechanics, supports the formulation of strategic approaches for both Alice and Bob, and enables the identification of key variables that govern the game's difficulty levels and strategic dynamics.

3.1. Random Graph Generation

It is important to begin with random graphs in order to observe how the game behaves under different structural conditions. Due to the nature of the majority rule, when a vertex with degree 1 is colored, its only neighbor being assigned the same color may cause the game to end immediately. This prevents the development of any meaningful strategy. For this reason, random graphs used in the simulations are generated with a minimum degree of 2 to ensure a valid and playable game setup.

In the simulations conducted within this study, random graphs are created using the *networkx* library in Python, specifically the *random_degree_sequence_graph* and *gnm_random_graph* functions. When generating graphs through a given degree sequence, degree-1 vertices are explicitly excluded to avoid trivial early terminations of the game.

In cases where the Erdős–Rényi model (*gnm_random_graph*) is used, the graph is filtered to ensure it is connected and that all nodes satisfy the minimum degree requirement. This additional step guarantees that both Alice and Bob encounter realistic and strategically meaningful positions throughout the game. Testing the game on connected random graphs makes it possible to evaluate strategy performance under less predictable structural conditions. Including such graphs in the experimental setup complements the analysis of well-defined families like star, cycle, or wheel graphs and helps assess whether particular heuristics work only in regular graphs or also in arbitrary networks.

Lastly, this approach forms a foundation for testing the resilience of the game strategies in adversarial scenarios, where Bob actively attempts to violate the majority rule under varying and uncertain structural configurations.

3.2. Odd Degree Vertex Configurations

Odd-degree vertices are considered structurally significant in the Majority Coloring Game, as they are more prone to triggering majority rule violations. To examine how both the number and positioning of such vertices affect game structure and complexity, graphs were generated with varying ratios and spatial distributions of odd-degree vertices. For this purpose, custom degree sequences were manually constructed to represent specific configurations such as graphs with a high concentration of odd-degree vertices or graphs where these vertices are more evenly distributed. These degree sequences were then transformed into valid simple graphs using the Havel–Hakimi algorithm [10], implemented through the NetworkX library. This method guarantees that the generated graphs conform to the specified degree constraints while maintaining simplicity and connectivity.

In addition to varying the number of odd-degree vertices, their spatial placement within the graph was systematically adjusted. After the initial construction using the Havel–Hakimi algorithm, rewiring operations were applied to relocate odd-degree vertices without changing the overall degree distribution. By varying the extent of rewiring, from more centralized to more dispersed layouts, graphs were generated that preserve the same degree distribution while differing only in the placement of odd-degree vertices. This controlled design enables a focused examination of how both the number and spatial distribution of structurally critical vertices affect gameplay dynamics, without introducing additional topological randomness that might arise from fully stochastic generation methods.

3.3. Graph Structure

Graph structures exhibit significant variation in their complexity, and as this complexity increases, devising a robust strategy and anticipating future moves becomes more difficult. Conversely, in graph families with different graph structures such as cycles, stars, or wheels, the game setup becomes more predictable, facilitating the application of predefined strategies.

Structures containing odd-degree vertices were given particular attention, as these elements introduce specific vulnerabilities. To ensure that the majority rule is preserved, Alice must avoid being the first to color an odd-degree vertex whose neighbors are still uncolored. Otherwise, Bob retains a clear opportunity to force a rule violation. For this reason, the number and positioning of odd-degree vertices were systematically varied during instance generation to examine their influence on game difficulty.

Furthermore, predefined graph families were employed to provide controlled environments for testing. These included well-known structures with varying degrees of symmetry and centrality. Additional instances were generated using custom-defined degree sequences, allowing for targeted modifications to structural properties while maintaining connectivity. Disconnected graphs or those with extreme degree values were not included in this study.

This controlled variation of graph characteristics supports a clearer assessment of how structural complexity impacts the feasibility of majority coloring strategies and overall gameplay dynamics. Together, these factors including structural complexity, odd-degree vertex configuration, and the choice of strategy form a comprehensive set of performance metrics.

3.4. Graph Size

Graph size is a fundamental parameter influencing the dynamics of the Majority Coloring Game. A larger graph imposes additional constraints on Alice, who must ensure that each vertex is colored in accordance with the majority rule, while also increasing the number of potential positions that Bob can exploit to induce violations.

An increase in graph size also raises the likelihood of encountering a higher number of odd-degree vertices, which can further complicate the coloring process and reduce Alice's chances of completing the game without a rule violation. Likewise, identifying structural patterns that support effective strategies becomes more difficult as the graph expands in size and complexity. In other words, as the graph grows, it becomes harder to identify and use local structures (such as cycles, hubs, or clusters) that can support effective strategies.

In the context of game design, graph size functions as a central variable in determining difficulty levels. Adjusting the number of vertices allows for controlled testing of how size impacts both strategy development and solution time. By varying graph size across instances, it becomes possible to systematically examine its effect on practical color usage and strategy performance.

3.5. Some Well-Known Graph Families

As part of the instance generation process, several predefined graph families were incorporated, including cycles, stars, wheels, the Petersen graph, and more complex vertex-transitive, fully odd-degree, 3-regular graphs such as the Cubical, Heawood, and Desargues graphs. These graph types were selected based on their well-defined topological characteristics and established relevance in graph coloring studies.

The graphs were generated using standard functions provided by the NetworkX library. For graph families with scalable structures namely cycles, stars, and wheels instances were created at multiple vertex counts to allow for size-dependent comparisons.

The Petersen, Cubical, Heawood, and Desargues graphs, due to their fixed structure, high symmetry, and vertex-transitivity, were included as representative non-scalable instances that exemplify more complex topologies.

The use of these graph families ensured structural consistency and reproducibility in the generation of test cases. Their regular degree sequences, symmetry, and well-understood properties make them suitable for examining algorithm behavior across diverse graph topologies. Furthermore, these predefined graphs serve as structured templates in the design of user interfaces, supporting the development of gameplay scenarios that reflect varying levels of complexity. In addition, using graph families with well-established theoretical bounds for their chromatic numbers and game coloring numbers provides a robust benchmark for comparing observed results with expected values.

4. Performance Metrics

In the design of the game, since Bob will be the computer-controlled player, distinct strategies will be developed for him to represent various difficulty levels. To evaluate the reliability and effectiveness of these strategies, a clear set of performance metrics will be defined. The determination of difficulty levels and strategic complexity will consider several key factors, including the structural and dimensional characteristics of the graph, the number of moves required by Bob to secure a win, the total number of colors used until the game concludes, and the statistical averages of these indicators.

Additionally, theoretical bounds such as the Majority Coloring Game chromatic number $\mu_g(G)$ and the majority game coloring number $\mathrm{col}_g(G)$ will be included as benchmarks, providing insight into the minimum and worst-case number of colors required under majority constraints and adversarial play. Together, these practical and theoretical metrics will serve as the basis for assessing the consistency, difficulty, and robustness of the proposed strategies across different game scenarios.

4.1. Number of Moves Required for a Win

During the game design process, various strategies were developed for both Alice and Bob to analyze the structural dynamics of the game and the logic underlying successful play. In this context, the total number of moves required by either player to secure a win has been treated as a key indicator in assessing the game's level of difficulty. In particular, the number of moves Bob needs to win has been used as a primary metric for evaluating difficulty across different game scenarios.

The game interface designed in this thesis positions the user as Alice and the computer as Bob. Accordingly, difficulty is assessed from the player's perspective, and the number of moves required by Bob to achieve victory functions as a central criterion in this evaluation framework.

When Bob needs to make a higher number of moves to win, it indicates that Alice's strategy is more robust, and the game is relatively easier for the player. Conversely, when Bob can win in fewer moves, it suggests that Alice is under greater strategic pressure, reflecting a higher level of difficulty. Thus, move count serves not only as a performance measure but also as an indicator of the balance of control and pressure between the two players.

This metric is used to compare performance across different graph types, vertex counts, and strategy combinations. By tracking the number of moves in diverse configurations, it becomes possible to systematically evaluate how structural complexity and strategy selection influence gameplay dynamics and difficulty levels.

4.2. Number of Colors Used

In this study, the number of colors used by Alice to complete the game without violating the majority rule is treated as a central indicator of difficulty. Instead of relying on theoretical bounds or fixed color limits, difficulty is evaluated through empirical data obtained from simulations involving different graph types, sizes, and strategies. The fewer colors Alice needs, the more efficiently she maintains the majority rule, indicating that the game is relatively easier for her to defend. Conversely, a higher number of colors implies increased structural or strategic pressure, raising the level of difficulty. In scenarios where Alice and Bob are given a fixed color set, this restriction further limits Alice's defensive flexibility and increases the challenge of avoiding majority rule violations.

The number of colors used by Alice may vary depending on structural features of the graph such as its level of centrality, symmetry, and vertex connections as well as the aggressiveness of Bob's strategy during the game. Recording the number of colors used in each run provides a practical measure for empirically comparing different strategies and graph structures. In this way, the level of game difficulty can be evaluated consistently without requiring separate formal analyses for each scenario.

Although this study does not aim to compute or prove the exact majority chromatic number $\mu_g(G)$, the number of colors used in successfully completed games serves as an empirical upper bound for this value. Each instance where Alice colors the entire graph without violating the majority rule demonstrates that $\mu_g(G)$ is less than or equal to the number of colors used. This empirical approach offers insight into the range of values that $\mu_g(G)$ may take under adversarial conditions and may serve as a reference point for future theoretical work.

5. Methodology

Understanding the dynamics of the Majority Coloring Game requires a close examination of the interactions between the two players, Alice and Bob, as well as an exploration of how various structural and strategic parameters influence the course and difficulty of the game. While chromatic and coloring numbers have been determined for certain graph families with specific properties, there is currently no exact method for computing such parameters across arbitrary graph types. Rather than focusing on the derivation of precise chromatic bounds, this study aims to explore how strategic behavior, structural characteristics, and rule constraints shape the gameplay experience.

In the context of this thesis, a variety of strategies were implemented for both Alice and Bob. Although Alice will ultimately be controlled by the user through the game interface, modeling her actions via a predefined algorithm allowed for a deeper investigation of her decision-making process, especially under adversarial pressure from Bob. For Bob, distinct strategies were designed to reflect varying difficulty levels and to systematically evaluate his ability to challenge Alice under different conditions.

Furthermore, because the game is designed to be played in real time through an interactive interface, players will not have full visibility of optimal moves. Therefore, the design process must also consider scenarios in which Alice is forced into a loss, as well as strategies that can help her avoid such outcomes. By analyzing how these scenarios arise and which parameters contribute to them, the study seeks to inform the design of game difficulty levels and support the development of adaptive strategy models. The steps of a heuristic algorithm based on a greedy approach, which colors all the vertices of a graph without violating the majority rule, are illustrated in Figure 5.1.

Input: Graph $G = (V, E)$ with vertices V and edges E
Output: Majority coloring $c(v)$ for all $v \in V$
Initialize:
    Colors $\leftarrow \{1\}$ {Start with one color}
    Majority[v] $\leftarrow \lceil \deg(v)/2 \rceil$, for all $v \in V$ {Majority threshold for each vertex}
    NeighborColorFrequency[u] $\leftarrow \emptyset$, for all $u \in V$ {Frequency of neighbor colors, initially empty}
while $V \neq \emptyset$ **do**
    Select a random vertex $v \in V$
    NeighborColors $\leftarrow$ NeighborColorFrequency[v]
    for each color $c \in$ Colors **do**
        if Frequency(c) $\leq$ Majority[v] **then**
            Assign $c(v) \leftarrow c$
            for each neighbor u of v **do**
                NeighborColorFrequency[u][c] $\leftarrow$ NeighborColorFrequency[u][c] + 1
            end for
            break
        end if
    end for
    if no valid color c found **then**
        Add new color c_{new} to Colors
        Assign $c(v) \leftarrow c_{new}$
        for each $u \in V$ **do**
            Initialize NeighborColorFrequency[u][c_{new}] $\leftarrow 0$
        end for
    end if
    Remove v from V
end while
Return $c(v)$ for all $v \in V$

Figure 5.1. Majority coloring heuristic algorithm.

The majority constraint ensures that each vertex v has a threshold defined as

$$\text{Majority}[v] = \left\lceil \frac{\deg(v)}{2} \right\rceil. \tag{5.1}$$

This guarantees compliance with the majority rule. The algorithm assigns colors locally by processing each vertex v independently while considering the colors of its neighbors to avoid any violations of the constraint. If the existing colors cannot satisfy the rule for a vertex, a new color is added to the palette. Following a greedy approach, the algorithm iteratively assigns colors while maintaining the constraints. This method can be implemented in any programming language to compute a majority coloring for a given graph.

5.1. Design of the Majority Coloring Game

At the core of the Majority Coloring Game design lies the assignment of effective strategies to Alice and Bob. The one unchanging rule of the game is that Alice aims to color the entire graph without violating the majority rule, while Bob's objective is to disrupt this rule. For Alice to win, the entire graph must be colored in accordance with the majority rule; however, for Bob to win, it is sufficient to violate the rule at just a single vertex.

Since Bob is typically the player making aggressive moves during the game, the strategies he develops to sabotage Alice's progress significantly increase the difficulty for Alice to succeed. Therefore, strategies developed for Alice should not only guide her toward a valid coloring of the graph, but also help her avoid potential sabotage by Bob as much as possible. However, considering that Bob needs only one vertex to violate the rule to win, while Alice must manage the majority rule for all vertices, Alice's strategy must adopt a broader and more holistic perspective.

During gameplay, when Alice is not under immediate threat, she should aim to make optimal moves that facilitate her path to victory. When she is under threat from Bob, she must shift to a defensive strategy. Bob, on the other hand, maintains an aggressive approach throughout the game, aiming to limit Alice's options and create situations that put her under constant threat of violating the majority rule.

Just like in a computer-based chess game, the game can be designed with varying difficulty levels. This allows Alice to have a fair chance of winning in easier levels, while also encouraging the development of more robust strategies as players become more experienced. In the easier levels, Alice can be given an advantage by programming Bob to ignore some of Alice's suboptimal moves, to avoid certain risky situations, or to focus on local rather than global strategies. In the more difficult levels, however, Bob should attempt to violate the majority rule under every possible condition, respond to each of Alice's moves with a new threat, and aim to win the game in as few moves as possible.

As the difficulty level increases, it is expected that Alice's chances of winning will decrease, and Bob will require fewer moves to secure a victory. Consequently, in the harder versions of the game, Alice will be required to develop much stronger strategies and more sophisticated defensive tactics to succeed.

5.1.1. The Most Basic Version of the Game (Greedy Strategy)

In the simplest version of the Majority Coloring Game, the fundamental rule remains unchanged: Alice aims to color the entire graph without violating the majority rule, while Bob's goal is to induce a violation at any vertex. However, both players employ relatively non-aggressive strategies in this version. Alice always plays first and adopts a defensive approach by prioritizing nodes with limited coloring options those considered "risky" and avoiding colors already used by their neighbors. This early risk mitigation limits Bob's opportunities to sabotage the coloring process.

Bob, on the other hand, does not engage in highly coordinated or systematically aggressive behavior. Instead, his strategy combines probabilistic and local sabotage. With a 20% probability, Bob makes a passive move by selecting a valid color without pursuing sabotage. With the remaining 80%, he attempts to apply localized pressure choosing nodes and colors that could lead to a future violation, especially where color dominance is already developing. This mixed approach was selected after testing showed that a 20–80 split provided the most balanced gameplay experience. Higher rates of passive behavior made Bob ineffective, while fully aggressive play caused the game to become overly difficult for Alice.

Additionally, Bob's node selections remain largely random in this version, which means he often fails to directly respond to Alice's most strategic moves. As a result, Alice can more easily preserve majority rule across the graph, especially when nodes with high majority thresholds are involved, further reducing Bob's ability to force a violation. A high majority threshold means that a vertex can have more neighbors of the same color without violating the rule, which reduces Bob's ability to force a violation with a single move and increases Alice's defensive flexibility. This design allows Alice to win in a reasonable number of games, making this version an effective baseline for understanding the game mechanics before introducing more advanced strategies and difficulty levels. The scenario in Figure 5.2 illustrates a case where the majority rule is not violated and Alice wins the game.

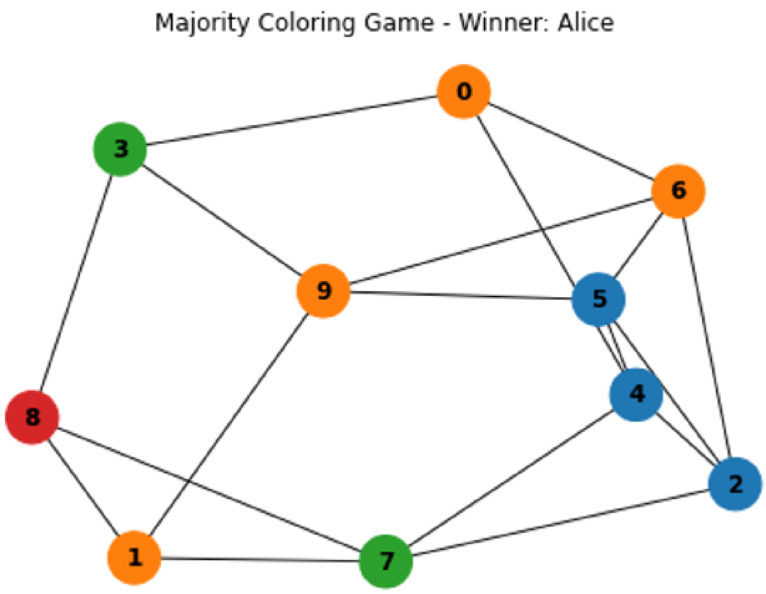


Figure 5.2. End of a game.

The moves in the game scenario are as follows:

(i) Alice colors vertex 0 with color 1 (orange).
(ii) Bob colors vertex 2 with color 2 (blue).
(iii) Alice colors vertex 1 with color 1 (orange).
(iv) Bob colors vertex 6 with color 1 (orange).
(v) Alice colors vertex 3 with color 3 (green).
(vi) Bob colors vertex 5 with color 2 (blue).
(vii) Alice colors vertex 7 with color 3 (green).
(viii) Bob colors vertex 4 with color 2 (blue).
(ix) Alice colors vertex 8 with color 4 (red).
(x) Bob colors vertex 9 with color 1 (orange).
(xi) Alice wins! The graph is fully colored.

As observed, although Alice began the game by coloring vertex 0, which has an odd degree and could have been exploited by Bob through targeted counter-moves, Bob instead opted to color a random vertex. This missed strategic opportunity allowed Alice to gain the upper hand and ultimately win the game. Despite the presence of numerous odd-degree vertices in the graph, offering Bob potential avenues for an easy victory, a more effective strategy on his part could have secured the outcome. His failure to respond systematically to specific threats significantly simplified the gameplay for Alice. To increase the complexity and competitiveness of the game, more advanced strategic mechanisms should be incorporated for Bob. The game structure corresponding to this greedy version is outlined below.

(i) Game Initialization
   - All nodes start uncolored.
   - Each node has a majority threshold, as defined in Equation (5.1), where the threshold is the ceiling of half the degree of the vertex.
   - The game begins with Alice and proceeds in alternating turns with Bob.

(ii) Alice's Strategy – Defensive Coloring
   - Alice prioritizes nodes with fewer valid coloring options.

- Among those, she tries to use an existing color first to minimize the number of colors used reflecting the fact that in the actual game Alice has a fixed set of colors to work with.
- If no existing color can be safely used, she tries a new color that won't violate any majority constraints.
- If no valid color is found for a node, Bob wins.

(iii) Bob's Strategy – Probabilistic Sabotage Coloring

- With a 20% probability, Bob simply picks any valid color without actively trying to create a violation; in this passive mode, he may even ignore some clear opportunities for sabotage and make a random move instead.
- With an 80% probability, Bob adopts a sabotage-oriented heuristic: he tries to select a color that could lead to a majority rule violation in neighboring nodes in the near future, targeting nodes where the chosen color is already dominant among neighbors, increasing the risk of a future violation. If no immediate sabotage move is possible during this mode, Bob chooses the next most disruptive option available, maximizing pressure on Alice's defense. If Bob successfully causes a majority rule violation at any point, the game ends immediately and Bob is declared the winner.

(iv) Game End Conditions

- Bob wins immediately if a color assignment causes a majority rule violation in any node.
- Alice wins if the entire graph is successfully colored without any such violation.

In this greedy version of the game, Bob's probability of making passive moves can be controlled as a factor that facilitates Alice's victory. If the goal is to make the game easier for Alice, this probability can be increased. However, for the game to remain fair, it would not be reasonable for all of Bob's moves to be random. Bob's aggressive moves will increase the difficulty of the game, and it should be analyzed which of his actions put more pressure on Alice and which are the most effective in breaking the majority rule.

5.1.2. Stackelberg Game Theory

During the design process of the game, a wide range of game theory models were thoroughly examined. These included cooperative game types such as bargaining, coalition, and matching games, as well as various non-cooperative models such as auction, Bayesian games, stochastic games, potential games, and queueing theory-based games. However, considering the turn-based, adversarial, and threat-driven nature of the Majority Coloring Game, Stackelberg games were identified as the most appropriate framework.

Cooperative models, which assume players act in coordination for mutual benefit, do not align with the structure of the Majority Coloring Game, where Alice and Bob operate with entirely opposing goals: Alice aims to color the entire graph without violating the majority rule, while Bob seeks to win by causing a single violation. For this reason, cooperative approaches are unsuitable for this setting. Uncertainty-based models such as Bayesian or stochastic games are designed for environments where players do not have complete information. However, in the Majority Coloring Game, all game information is fully observable to both players, making such models less applicable [11].

Stackelberg games are defined in the literature as sequential decision-making models based on a leader–follower relationship [12]. In these models, one player acts as the leader and makes the first move, while the follower observes the leader's action and optimizes their own strategy accordingly. This structure is widely used to analyze various strategic interactions ranging from economic competition to defense planning. Although classic Stackelberg models assume fixed roles, the Majority Coloring Game requires dynamic role changes. For instance, Bob may behave as an aggressive leader when Alice is forced into defense, while in other scenarios, Alice may anticipate threats and adopt a proactive strategy. This fluid role transition highlights the need for flexibility in the model and led to the adoption of a dynamic Stackelberg approach in the game's logic.

In conclusion, the Stackelberg model was chosen as the foundational strategic framework for the game design of the Majority Coloring Game due to its suitability for modeling the game's competitive structure, its ability to support alternating leader–follower roles, and its capacity to represent both offensive pressure and defensive tactics in a realistic way. The leader-follower roles of Alice and Bob can be defined as in Table 5.1.

Table 5.1. Leader-follower roles of Alice and Bob.

Player	**Role**	**When**	**Strategy**
Alice	Leader	When no risky nodes are present	Selects a random uncolored node, avoids using the same color as neighbors
Alice	Follower	When risky nodes are present	Selects a risky node, explicitly avoids using the same color as neighbors
Bob	Leader	When he makes a move that directly violates the majority rule, or sets up a position that will cause a violation in the next move	Attempts risky coloring; if he can violate the majority rule, he wins
Bob	Follower	When he is not actively forcing an immediate violation and only responds to Alice's moves	Tries to color neighbors of Alice's nodes with the same color to apply pressure

In the Stackelberg algorithm developed within this game structure, a risky node is any uncolored vertex that Bob could exploit to cause an immediate majority rule violation on his next move. Identifying and defending these nodes helps Alice maintain the majority rule. Below are Alice's and Bob's strategies in more detail:

(i) Alice is Leader (When no risky nodes exist)
- Alice selects a random uncolored node.
- If the selected node has already-colored neighbors, she avoids using their colors.
- If no valid color is available, she creates a new color and adds it to the color set.
- The aim is to prevent any future majority rule violation.

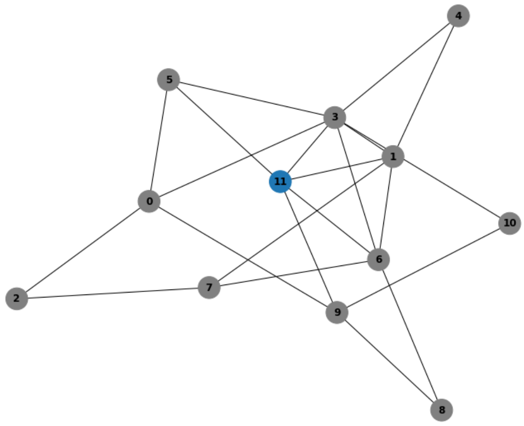
(ii) Alice is Follower (When risky nodes exist)
- Alice uses the *bob_can_win()* function to identify risky nodes.
- She selects one of these risky nodes and colors it to block Bob's threat.
- While coloring, she avoids using colors that are already dominant among its neighbors to reduce future risks.

(iii) Bob is Leader (When he can immediately force a majority rule violation or set up a position for it in the next move)
- Bob checks all possible color assignments to see if he can either directly break the majority rule or create a position that guarantees a violation on his next turn.
- If so, he makes that move immediately securing a win now or setting up an inevitable win.

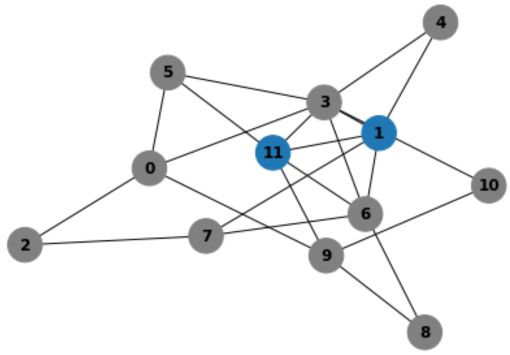
(iv) Bob is Follower (When he cannot immediately force or set up a violation)
- Bob applies strategic pressure by selecting nodes and colors that increase the risk of a future majority rule violation (e.g., creating local color dominance).
- If no effective pressure move is possible, Bob makes a passive move by randomly selecting a valid node and color.

An example of the game and all corresponding moves are illustrated sequentially in Figure 5.3. During each move, the leader-follower roles of Alice and Bob dynamically shift as follows:

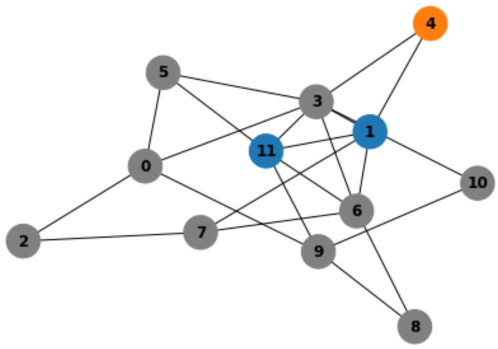
(i) Alice (leader): Alice colored node 11 with color 0 (blue)
(ii) Bob (follower): Bob strategically colored node 1 same as neighbor 11 with color 0 (blue)
(iii) Alice (leader): Alice colored node 4 with color 1 (orange)
(iv) Bob (follower): Bob strategically colored node 3 same as neighbor 1 with color 0 (blue)
(v) Alice (follower): Alice colored node 5 with color 1 (orange)
(vi) Bob (leader): Bob colored node 6 with color 0 (blue) and violated the majority rule. Bob wins!



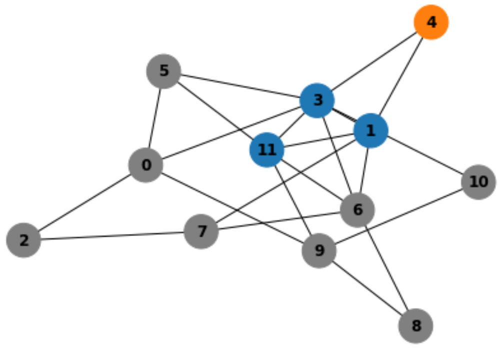
(a) Alice's first move.



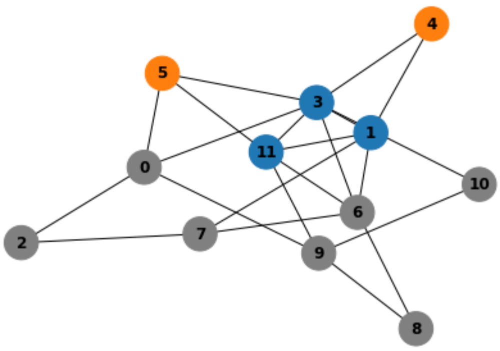
(b) Bob's first move.



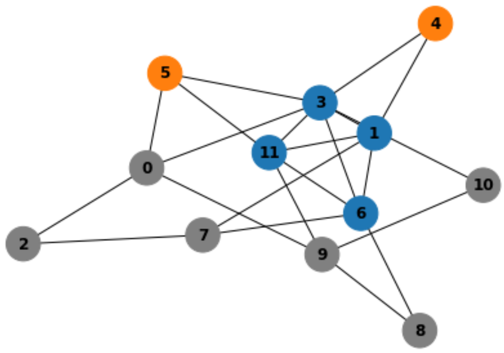
(c) Alice's second move.



(d) Bob's second move.



(e) Alice's third move.



(f) Bob's third move.

Figure 5.3. Example game scenario.

5.1.3. Odd Degree Vertices

Odd-degree vertices play a critical role in shaping the dynamics of the Majority Coloring Game. As illustrated in Figure 5.4, a sequence of moves demonstrates how an early decision around such a vertex can lead to a loss. In this scenario, Alice begins by coloring vertex 1 with pink. Bob immediately responds by coloring vertex 2 with the same color. Although Alice defensively colors vertices 3 and 5 with a different color (blue), Bob continues by coloring vertices 4 and 6 with pink, successfully triggering a majority rule violation and winning the game. This example shows how vulnerable odd-degree vertices can be exploited, especially when surrounded by uncolored neighbors.

One of Bob's core strategies is to identify such odd-degree vertices and steer the game in a way that turns them into potential risky nodes for Alice. This forces Alice to color these nodes under pressure to prevent a majority rule violation. As the number of odd-degree vertices increases, Bob's chances of creating risky situations and successfully sabotaging Alice's strategy grow, thereby raising the overall difficulty level of the game.

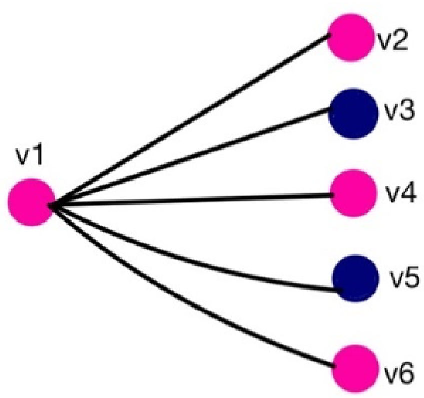


Figure 5.4. Odd degree vertex scenario.

Beyond their quantity, the structural positioning of odd-degree vertices significantly affects gameplay. When such vertices are distributed across central or peripheral regions of the graph, the defensive burden on Alice shifts. Peripheral distributions may expose more entry points for Bob, while central clustering allows him to pressure the graph's core. Thus, not only the number but also the strategic placement of these vertices must be considered when evaluating game complexity. While constructing the graphs with odd degree vertices specifically, the goal was to examine the impact of the graph structure on the Majority Coloring Game. To determine whether the odd-degree vertices in a graph influence the strategies of Alice and Bob, the game can be played on a graph where the vertex degrees are manually defined, rather than on a randomly generated graph with a random degree sequence. Defining the degree sequence manually is important for modifying the structure of the graph and evaluating the related outcomes.

In this section of the thesis, the Havel-Hakimi algorithm is used. The Havel-Hakimi algorithm, developed independently by Havel and Hakimi, is a method for determining whether a given degree sequence corresponds to a simple graph and, if so, for constructing such a graph [10]. The algorithm operates by iteratively selecting the vertex with the highest degree, connecting it to the next highest-degree vertices, and updating the degree sequence accordingly. This process continues until all degrees are reduced to zero, confirming that the sequence is graphical. If this condition is not satisfied at any point, it implies that no such simple graph exists. As an alternative to the Erdős-Gallai theorem, this algorithm provides a practical approach to generating graphs with specific degree sequences.

With a time complexity of $O(n \log n)$ due to the initial sorting step, the Havel–Hakimi method is especially useful for generating graphs that match a given degree distribution, which is valuable in network simulations, for example when modeling biological systems while preserving degree constraints [10]. The main reason for choosing the Havel–Hakimi algorithm in this study is its ability to both test whether a given degree sequence is graphical and construct a corresponding simple and connected graph in practice. This enables controlled adjustment of parameters such as the number and placement of odd-degree vertices, as well as minimum degree constraints, which would not be reliably achievable with fully stochastic models like Erdős–Rényi. To systematically test the effect of the odd degree vertices through the Stackelberg algorithm, which serves as the game's core strategy model, graphs with predefined degree sequences were generated using the Havel–Hakimi method. In the simulation designed for the game, a built-in library for this method was utilized. As a result, the degree sequences of the randomly generated graphs for the Majority Coloring Game can be configured before the game begins. In Figure 5.5, we can observe the final state of a game played on a graph with the degree sequence $(3, 3, 4, 6, 5, 2, 7, 5, 3)$. The moves are as follows:

(i) Alice colors node 4 with color 1 (orange).
(ii) Bob colors node 7 with color 1 (orange).
(iii) Alice colors node 2 with color 1 (orange).
(iv) Bob colors node 3 with color 1 (orange).
(v) Majority rule violated. Bob wins!

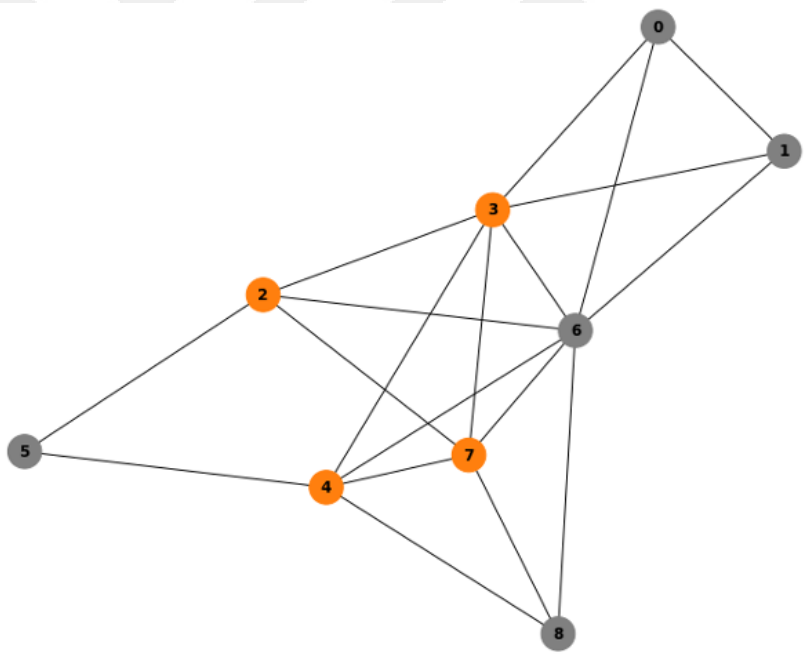


Figure 5.5. Odd degree vertex and end of a game.

During gameplay, it is known that Alice should avoid coloring odd-degree vertices at the beginning; otherwise, Bob can easily develop a winning strategy. If Alice is forced to color an odd-degree vertex, Bob may subsequently trap her and cause a violation of the majority rule. Therefore, both the number and the spatial placement of odd-degree vertices that is, their positions within the visual layout of the graph are considered critical factors in the strategies developed by both Alice and Bob, as well as in determining the overall difficulty of the game. As demonstrated in the example game in Figure 5.5 , Alice's decision to begin by coloring an odd-degree vertex and her reluctance to introduce a new color ultimately made it impossible for her to win the game.

5.1.3.1. Alice's strategy for avoiding odd degree vertices. It is well-established that Alice should avoid coloring odd-degree vertices during the early stages of the game, or in situations where she is not required to counter a critical move initiated by Bob. Extending this reasoning, Alice should also refrain from coloring any uncolored vertex that possesses an odd number of uncolored neighbors at any point throughout the game, as such configurations are more susceptible to violations of the majority rule. To mitigate this risk, a strategic approach for Alice would involve preemptively coloring the uncolored neighbors of these vertices in a manner that reduces their number of uncolored adjacent vertices to an even value, thereby minimizing potential threats before proceeding to color them.

To evaluate the effectiveness of this strategy, a series of experiments can be conducted in which Alice either follows or disregards the approach of prioritizing the reduction of uncolored odd-degree vertices. The comparative outcomes of these games, such as the number of moves before a majority rule violation occurs, can offer insights into the practical utility of the strategy. Preliminary observations suggest that by systematically converting odd-degree vertices into even-degree ones through targeted neighbor coloring, Alice is able to delay majority violations, thereby increasing her chances of successfully completing the coloring process.

A critical vertex is defined as a vertex that, if not colored correctly by Alice, will result in a violation of the majority rule. At any point during the game, if a critical vertex exists, Alice is obliged to use her move to color it. Otherwise, Bob will be able to color this vertex on his next turn in a way that violates the majority rule, thereby winning the game. If multiple critical vertices exist simultaneously and Bob is aware of them, he is guaranteed to win the game.

A safe vertex, on the other hand, refers to a vertex with an even number of uncolored neighbors. These vertices are considered risk-free for Alice, as coloring them will not lead to a majority rule violation. In the absence of any critical vertices and without applying a specific strategy, Alice may prioritize coloring safe vertices.

Alice may apply the strategy of reducing the number of odd-degree vertices only when there are no critical vertices present during her turn. In the algorithm developed for this strategy, when Alice's turn comes, she first checks for the presence of any critical nodes. If none are found, she identifies the odd-degree vertices and selects one of their neighbors to color in such a way that their number of uncolored neighbors becomes even. When selecting which vertex to color, the algorithm aims to choose the one that leads to the greatest reduction in the total number of odd-degree vertices in the graph. Alice's Odd Degree Reduction strategy, as outlined above, can be summarized in the following four steps:

(i) Critical Vertex Check
   - Determine whether there exists a vertex that Bob could use in his next move to violate the majority rule.
   - If such a vertex is found, Alice prioritizes coloring it with a proper color.This step serves a defensive purpose, aiming to prevent immediate loss.

(ii) Odd-Uncolored Neighbor Reduction Strategy
   - If no critical vertex exists: Alice analyzes which vertex, if colored, would most effectively reduce the number of neighboring vertices with an odd number of uncolored adjacent nodes, thereby minimizing potential majority rule risks. This is intended to mitigate future risks.
   - The objective is to balance the local structure, thereby weakening Bob's potential to create critical threats.

(iii) Safe Vertex Selection
   - If neither critical nor structurally advantageous vertices are available: Alice prioritizes vertices with an even number of uncolored neighbors (i.e., safe vertices).

- These are statistically less likely to lead to a majority rule violation.

(iv) Random Selection as a Last Resort

- If none of the three strategies above are applicable: Alice proceeds by selecting a vertex at random.

Meanwhile, Bob follows this strategy:

(i) Critical Node Exploitation

- Bob constantly monitors for any nodes that, if colored now or soon, could violate the majority rule.
- If such a node exists, he immediately colors it and wins the game.

(ii) Pressure Moves

- Bob applies local pressure by coloring the neighbors of Alice's nodes with the same color.
- This reduces Alice's safe options and increases the risk of a future majority rule violation.

(iii) Passive Move (Alternative)

- If no immediate or strategic sabotage is possible, Bob makes a passive move by randomly selecting a valid node and color.
- This keeps the game moving while he waits for new opportunities to emerge.

In Figure 5.6, the final state of a game in which the Odd Degree Reduction strategy was applied is presented. The sequence of moves made throughout the game is listed below. Since there is no critical vertex at the beginning of the game, Alice, in her first move, selects the vertex that most significantly reduces the number of vertices with an odd number of uncolored neighbors. The scanning process she performs to identify this vertex is as follows:

(i) Alice evaluating node 0: odd_before=4, odd_after=4, reduction=0

(ii) Alice evaluating node 1: odd_before=3, odd_after=1, reduction=2

(iii) Alice evaluating node 2: odd_before=1, odd_after=1, reduction=0

(iv) Alice evaluating node 3: odd_before=4, odd_after=5, reduction=-1

(v) Alice evaluating node 4: odd_before=2, odd_after=4, reduction=-2

(vi) Alice evaluating node 5: odd_before=1, odd_after=1, reduction=0

(vii) Alice evaluating node 6: odd_before=2, odd_after=2, reduction=0

(viii) Alice evaluating node 7: odd_before=3, odd_after=4, reduction=-1

(ix) Alice evaluating node 8: odd_before=3, odd_after=3, reduction=0

(x) Alice evaluating node 9: odd_before=4, odd_after=0, reduction=4

(xi) Alice evaluating node 10: odd_before=2, odd_after=3, reduction=-1

(xii) Alice evaluating node 11: odd_before=1, odd_after=4, reduction=-3

(xiii) Alice evaluating node 12: odd_before=2, odd_after=1, reduction=1

(xiv) Alice evaluating node 13: odd_before=2, odd_after=4, reduction=-2

(xv) Alice evaluating node 14: odd_before=0, odd_after=5, reduction=-5

(xvi) Alice uses odd-uncolored reduction strategy! Picks vertex 9. Alice colors vertex 9 with color 0 (blue).

Subsequently, Alice and Bob take turns making their moves, with Alice continuing to apply her strategy until a critical vertex emerges. After Alice colors vertex 9 with color 0 (blue), the following moves are made in order: Bob and Alice color vertices 11, 2, 1, 5, and 7, respectively. When Alice's turn comes again, a critical vertex has appeared, and she is required to color vertex 12 with the correct color. Alice colors vertex 12 with color 1 (orange); however, at that point, multiple critical vertices exist one of which is vertex 4. In his final move, Bob colors vertex 4 with color 0 (blue), thereby successfully violating the majority rule and winning the game.

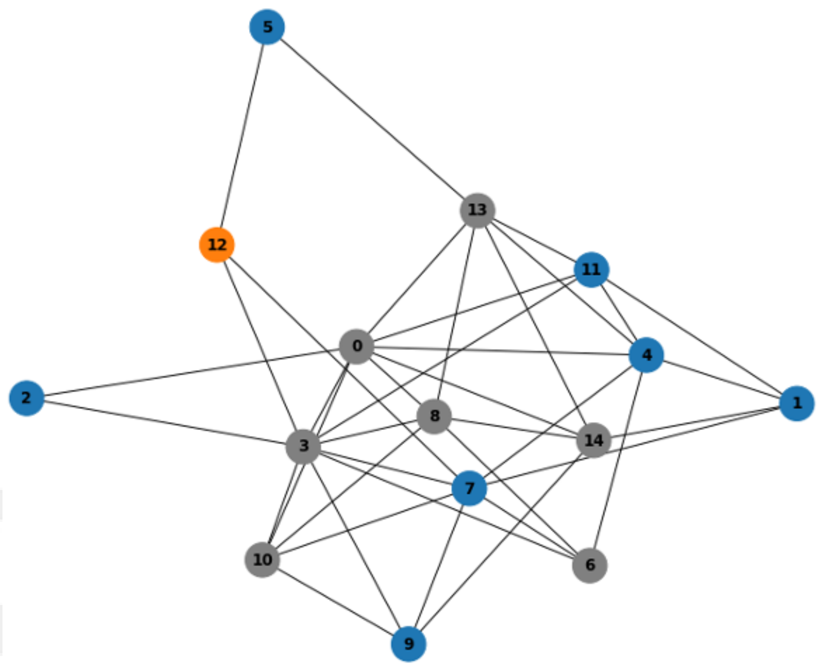


Figure 5.6. End of a game.

Odd-degree reduction is a defensive strategy that helps reduce risk but does not offer a complete guarantee. If Bob can directly color a critical vertex v, this mechanism limits Alice's ability to maintain control solely through her neighboring nodes. By taking control of the target vertex, Bob can exert color pressure on its neighbors and force a majority rule violation. This scenario shows that Alice's strategy of protecting only through neighbor coloring may sometimes be insufficient. However, the odd-degree reduction strategy does narrow Bob's potential threat space, limits the pressure he can apply through neighbors, and in most cases delays possible losses, giving Alice more time to defend. Therefore, while not foolproof on its own, it remains a valuable risk-reducing tactic that complements other defensive measures.

6. Experimental Results

The strategies explained in detail in the previous chapter are comprehensively evaluated in this section in terms of their difficulty levels, in light of the experimental findings. In particular, this section examines how changes in fundamental parameters such as graph size, the proportion and distribution of odd-degree vertices, structural symmetry, and color constraints significantly transform the dynamics of the Majority Coloring Game. The discussion elaborates on the conditions under which different graph topologies and Bob's adversarial strategies make it more challenging or easier for Alice to comply with the majority rule. In doing so, this section builds a bridge between theoretical upper bounds (such as the majority chromatic number and the majority game coloring number) and the experimental results, offering insights into how these parameters should be interpreted in practical gameplay scenarios. The findings provide a valuable contribution by demonstrating how the interaction between structural properties of graphs and gameplay constraints determines the strategic complexity and feasibility of majority coloring.

6.1. The Most Basic Version (Greedy Strategy)

In the simplest version of the game, Alice can easily color the entire graph in accordance with the majority rule. Bob does not act aggressively in his moves, meaning he does not specifically target Alice's future moves, and instead makes either strategic or random moves depending on the current state of the graph.

In the game, different values for Bob's probability of making random moves were tested, and Alice's win rate was analyzed accordingly. A 20% probability was selected, as it allows the game to proceed within a logical strategic framework while also creating a scenario where Bob is not overly aggressive, thus enabling the possibility of an Alice victory. Table 6.1 reports the outcomes of simulations conducted under different probability settings, which determine the likelihood of Bob performing random rather than strategic moves. All experiments were carried out on a random connected graph consisting of 20 vertices. To ensure comparability across conditions, the same graph structure was used for each probability level, with 10 independent simulation runs per setting. This controlled design enables an isolated examination of how varying degrees of randomness in Bob's behavior influence the difficulty and outcome of the game.

Table 6.1. Win rates by randomness level in Bob's strategy.

Bob's Random Move Probability	**Alice Wins**	**Bob Wins**
0%	0	10
5%	4	6
10%	8	2
20%	9	1
30%	10	0
50%	10	0

Naturally, the number of moves required for the game to end in which Alice wins equals the number of vertices in the graph, because it is not possible for Alice to win without coloring the entire graph according to the rule. In this experimental setup, Alice always wins the game. Table 6.2 shows the change in the number of colors used in the game as the number of vertices changes.

Table 6.2. Graph size vs. coloring behavior in the majority coloring game.

Vertex Count	**AVG Total Moves**	**Avg Colors Used**
10	10	4.2
16	16	5.8
30	30	8.0
50	50	11.4

In this experiment, graphs with 10, 16, 30, and 50 vertices were used, with five runs carried out for each graph size. Five distinct random connected graphs were generated for each graph size, and for each graph instance, five independent simulation runs were conducted, resulting in a total of 25 runs per graph size.

Although there is no established formula for determining the chromatic number in the Majority Coloring Game, it is known that such graphs are 2-colorable under standard majority rule when no strategic play is involved. However, in a competitive setting where Alice and Bob take turns, the number of colors used tends to increase due to strategic constraints.

As the graph size increases, the number of colors used also rises but the change is not as dramatic as one might expect. For example, when the number of vertices increases from 10 to 50, the number of colors used increases from an average of 4.2 to 11.4. This growth remains relatively moderate given the fivefold increase in graph size. This is largely because Bob does not employ aggressive or complex sabotage strategies in this version of the game. While he avoids invalid or reckless moves, he also does not actively counter Alice's color expansions. As a result, Alice retains a high degree of control throughout the game, especially in larger graphs where she has more room to maneuver and avoid high-risk nodes.

The findings suggest that increasing graph size does not necessarily lead to a proportional increase in the coloring difficulty for Alice, provided Bob's behavior remains passive. In fact, larger graphs may even offer Alice more flexibility in distributing her moves and maintaining the majority rule without triggering violations. However, it should be noted that a higher number of colors used by Alice indicates increased effort and reflects that the game becomes more challenging for her to manage strategically. This observation applies specifically to scenarios where neither Alice nor Bob are restricted in the number of colors they can use.

The values shown in Table 6.2 present the game results of Alice and Bob in the Greedy strategy. In this game strategy, minimizing the number of colors used by Alice and Bob is not a primary objective. As an alternative version of the game, a rule was introduced where both players avoid using new colors unless necessary, and the outcomes were analyzed accordingly. The results of the game played with a strategy aimed at minimizing the number of colors used with the same experimental set-up are presented in Table 6.3.

Table 6.3. Game outcomes under the alternative greedy strategy.

Vertex Count	**AVG Total Moves**	**AVG Colors Used**
10	10	2.6
16	16	3
30	30	3
50	50	3

The fundamental difference between the two versions lies in their approach to using new colors. In the first version, players (especially Alice) did not hesitate to use new colors when necessary to avoid matching the colors of neighboring nodes, which noticeably increased color diversity. In contrast, the updated version adopts a stricter strategy that prohibits the use of new colors unless coloring with existing ones is not possible. This distinction caused the average number of colors used to rise as high as 11.4 in the first version, whereas in the updated version, it remained consistently around 3 across all graph sizes.

In both cases, Alice won all games, which can be attributed to Bob's strategy not being sufficiently aggressive. However, by imposing a color limit in the updated version, the number of colors used by Alice in graphs with a high number of vertices came closer to the theoretical upper bound of the majority chromatic number. This demonstrates a significant improvement in terms of strategic simplicity and efficiency. As a result, planning with a color constraint during gameplay helps Alice keep her defense more compact and effective, preventing unnecessary color diversity and reinforcing the overall robustness of her strategy.

6.2. Stackelberg Strategy

The main strategy used in the game's design, which forms the basis of the moderate and hard versions, is the Stackelberg strategy, where Alice and Bob dynamically switch between the roles of leader and follower. Table 6.4 presents the end-of-game data for games played on random graphs with different numbers of vertices.

In the harder versions of the game, where the Stackelberg strategy is employed, Bob adopts an aggressive approach in each run. Bob's goal is to put constant pressure on Alice's defense through each move, forcing Alice into defensive positions around risky structures (such as odd-degree vertices), thereby increasing her likelihood of making mistakes. This approach, especially when Bob acts as the leader, can compel Alice to color risky nodes under pressure, often leading to Bob winning the game.

For each vertex count, five different random connected graphs were used with five independent simulation runs each. The average number of moves and colors used were calculated to evaluate the impact of graph size on gameplay dynamics. Additionally, to better reflect the actual game scenarios in the designed interface, the strategy also focuses on minimizing the number of colors used whenever possible. Table 6.4 presents the results of a representative run for each graph size.

Table 6.4. Game outcomes under the Stackelberg strategy.

Run	**Vertex Count**	**Total Moves**	**Colors A**	**Colors B**	**Total Colors**
Run 1	10	5	2	1	2
Run 2	10	4	3	2	3
Run 3	10	4	2	1	2
Run 4	10	6	3	2	3
Run 5	10	5	2	1	2
Run 1	16	6	2	1	2
Run 2	16	8	2	1	2
Run 3	16	8	2	1	2
Run 4	16	8	3	2	3
Run 5	16	10	2	1	2
Run 1	30	12	5	3	5
Run 2	30	10	3	1	3
Run 3	30	12	3	2	3
Run 4	30	12	5	3	3
Run 5	30	14	3	2	3
Run 1	50	20	4	1	4
Run 2	50	26	4	2	4
Run 3	50	18	5	3	5
Run 4	50	20	5	2	5
Run 5	50	20	5	2	5

Table 6.5 summarizes the average outcomes across all simulation runs. As the number of vertices increases, both the total number of moves and the number of colors used tend to rise, reflecting increased complexity in larger graphs. In particular, the number of colors used by Alice rises significantly with larger graphs. This suggests that Alice needs more strategic flexibility and thus more colors to successfully color the graph without violating the majority rule.

The number of colors used by Bob, on the other hand, remains more limited and relatively stable throughout the experiments. This reflects Bob's reactive role in the game, where he focuses more on exploiting opportunities created by Alice's moves rather than expanding the color set himself. The fact that the total number of colors used closely aligns with the number of colors introduced by Alice further supports the idea that new colors are predominantly introduced by her. Alice tends to introduce new colors defensively to prevent Bob from potentially violating the majority rule.

The number of moves also increases proportionally with the graph size, indicating that the game lasts longer and becomes more strategically demanding on larger graphs. As the game progresses, it becomes increasingly difficult for Alice to make the right moves to block Bob's threats, while Bob finds more opportunities to pressure Alice into risky positions. On the other hand, the fact that the number of moves Bob needs to win also increases shows that the game becomes more challenging not only for Alice but also for Bob, as he must sustain his threats through longer and more carefully planned sequences of moves. Consequently, based on the average outcomes, Bob consistently applies pressure and succeeds in winning all of the games.

Table 6.5. Average move and color usage statistics under the Stackelberg strategy.

Vertex Count	**AVG Moves**	**AVG Alice Colors**	**AVG Bob Colors**	**AVG Colors**
10	4.8	2.4	1.4	2.4
16	8	2.2	1.2	2.2
30	12.0	3.8	2.2	3.8
50	20.8	4.6	2.0	4.6

In summary, increasing the graph size makes the Stackelberg strategy applied Majority Coloring Game more complex and challenging in terms of both move count and color management. Alice faces greater difficulty in maintaining compliance with the majority rule, while Bob has more room to create threats yet he must also work harder to convert these opportunities into a win.

6.3. Odd Degree Effects

The strategic importance of odd-degree vertices in the Majority Coloring Game is well known. Alice avoiding coloring odd-degree vertices, and Bob trying to force Alice into coloring them, is a strategy that increases the game's difficulty. To understand the significance of odd-degree vertices for the game, an experimental setup was established. For the graphs with 16, 32, 40, and 52 vertices, the effect of different proportions of odd-degree vertices on the game outcome was observed.

In this setup, an aggressive Stackelberg strategy was employed in which Bob maintains continuous pressure on Alice to create opportunities for majority rule violations. This strategy does not explicitly target odd-degree vertices but may force Alice into risky positions, including coloring such vertices, as part of the overall threat dynamics. In this way, insights could be gained into the relationship between the proportion of odd-degree vertices and the game's difficulty, based on the number of moves Bob needed to secure a win.

In the established setup, five distinct connected graphs were generated for each vertex count and odd-degree vertex ratio. For each graph, five independent simulation runs were performed. The average number of moves calculated from the games in which Bob successfully violated the majority rule was used to assess the overall difficulty.

As shown in Table 6.6, as the proportion of odd-degree vertices increases for graphs of different sizes, the number of moves Bob needs to win decreases. A reduction in the number of moves required for Bob to win indicates that the game becomes easier for Bob and more difficult for Alice. As the number of odd-degree vertices increases, even if Alice tries to avoid coloring these vertices, Bob is more frequently able to force her into doing so, allowing her to maintain the majority rule for a shorter time.

Table 6.6. Effect of odd-degree vertex ratio on game difficulty.

Setup	**Vertex Count**	**% Odd Degree**	**% Even Degree**	**AVG Moves**
Setup 1	16	0	100	8.4
Setup 1	16	25	75	8.4
Setup 1	16	50	50	4.4
Setup 1	16	75	25	5.6
Setup 1	16	100	0	4.4
Setup 2	32	0	100	15.2
Setup 2	32	25	75	9.6
Setup 2	32	50	50	6.4
Setup 2	32	75	25	4.4
Setup 2	32	100	0	4.8
Setup 3	40	0	100	21.6
Setup 3	40	25	75	15.6
Setup 3	40	50	50	12.8
Setup 3	40	75	25	12.8
Setup 3	40	100	0	10.4
Setup 4	52	0	100	31.2
Setup 4	52	25	75	20.8
Setup 4	52	50	50	15.6
Setup 4	52	75	25	14.0
Setup 4	52	100	0	10.8

At the same time, the relationship between changes in graph size and the number of moves in the game can also be examined. As observed, the number of moves in the game noticeably increases as the graph size grows. It indicates that as graph size increases in scenarios with a high proportion of odd-degree vertices, Alice's chances of reaching safer nodes improve. However, the expansion of potential threat areas makes it harder for her to maintain strategic control and defend against all of Bob's moves. At the same time, Bob gains more opportunities to create threats across different regions, but converting these opportunities into a quick majority rule violation becomes more demanding. This is because he must coordinate more complex sabotage actions and requires additional moves to effectively force a violation. As a result, the difficulty increases for both players in different ways: for Alice, the challenge lies in monitoring a wider network and blocking multiple threats, while for Bob, it lies in exploiting dispersed vulnerabilities and sustaining effective pressure. This section investigates not only the number of odd-degree vertices in a graph, but also how their spatial distribution affects game dynamics and difficulty levels. The analysis complements the previously defined performance metric based on the average number of moves required by Bob to win the game, which serves as a proxy for assessing the effectiveness of Alice's defensive strategies. The simulations are conducted within the Stackelberg strategy framework, in which Alice aims to anticipate and block Bob's potential majority rule violations.

To isolate the effect of vertex positioning, the same degree sequence was used for each graph size, while rewiring operations were applied after graph generation to manipulate the locations of odd-degree vertices without altering their degrees. These operations allowed for systematic control over the spatial layout while preserving the structural properties of the graph.

For graph construction, the Havel–Hakimi algorithm was employed to generate valid and simple graphs corresponding to the predefined degree sequences. This algorithm ensures that the resulting graphs contain no loops or multiple edges, thereby satisfying the criteria for a simple graph. As a result, a consistent structural baseline was established across all instances, allowing for the isolated examination of odd-degree vertex placement. The configurations were categorized as follows:

- Centralized: Odd-degree vertices are clustered in a single region.
- Moderate Shuffle: Vertex positions are partially randomized.
- Heavy Shuffle: Odd-degree vertices are widely dispersed throughout the graph.

Experiments were conducted using four different graph sizes (16, 32, 40, and 52 vertices) each with a fixed number of odd-degree vertices. For each configuration, a single connected graph was generated and used across five independent simulation runs. The average number of moves required by Bob to successfully violate the majority rule was recorded as the primary indicator of game difficulty.

Table 6.7. Effect of odd-degree vertex positioning on Bob's average winning moves.

Vertex Count	**Configuration**	**AVG Bob Moves to Win**
16	Centralized	4.4
16	Moderate Shuffle	4.8
16	Heavy Shuffle	4.0
32	Centralized	6.8
32	Moderate Shuffle	6.0
32	Heavy Shuffle	5.2
40	Centralized	10.8
40	Moderate Shuffle	6.0
40	Heavy Shuffle	5.0
52	Centralized	12.2
52	Moderate Shuffle	8.6
52	Heavy Shuffle	6.4
40	Centralized	12.4
40	Moderate Shuffle	8.4
40	Heavy Shuffle	6.8
52	Centralized	16.4
52	Moderate Shuffle	10.6
52	Heavy Shuffle	10.0

The results in Table 6.7 indicate that the spatial distribution of odd-degree vertices is a significant factor in shaping the course of the game. In particular, in centralized configurations where odd-degree vertices are clustered in a specific region Bob required more moves on average to secure a win. This suggests that Alice is better able to organize her defensive strategy when threats are concentrated in a limited area. In contrast, in heavy shuffle scenarios where the odd-degree vertices are widely dispersed, Bob's path to victory became noticeably shorter, and Alice's ability to defend effectively was reduced. This effect becomes even more pronounced as the graph size increases. For example, in 52-node graphs, the average number of moves Bob needed to win was 16.4 in the centralized configuration but dropped to 10 in the heavy shuffle scenario. This difference shows that when threats are concentrated in a single area, Alice is better able to organize her defensive strategy, whereas when threats are spread out, Bob can exploit vulnerabilities more efficiently. Ultimately, these findings highlight that analyses of game difficulty should consider not only the number of odd-degree vertices but also their spatial distribution within the graph.

6.3.1. Alice's Odd Degree Reduction Strategy

Previous sections have shown that the presence of odd-degree vertices has a substantial impact on the structure and difficulty of the Majority Coloring Game. In this section, a possible strategy that Alice can adopt to counter this structural challenge is examined. The strategy is based on identifying odd-degree vertices in the current game state and coloring the vertex that most reduces their number after each of Alice's moves. The underlying assumption is that by minimizing the number of odd-degree vertices, Alice can reduce the structural weaknesses that Bob might exploit. To evaluate the effectiveness of this approach, simulations were conducted on graphs with 10, 16, 30, and 50 vertices. For each graph size, a single random connected graph was generated and used for five independent simulation runs. The analysis focuses on the extent to which this strategy affects Bob's ability to win and its implications for the overall difficulty of the game.

Table 6.8. Game outcomes and color usage under Alice's odd degree reduction strategy.

Runs	Vertex Count	Total Moves	Wins	Colors A	Colors B	Total Colors
Run 1	10	8	Bob	2	1	2
Run 2	10	6	Bob	2	1	2
Run 3	10	10	Alice	3	2	3
Run 4	10	10	Alice	3	1	3
Run 5	10	8	Bob	2	1	2
Run 1	16	16	Bob	4	2	4
Run 2	16	14	Bob	3	1	3
Run 3	16	8	Bob	3	1	3
Run 4	16	16	Alice	4	1	4
Run 5	16	16	Alice	4	2	4
Run 1	30	16	Bob	5	1	5
Run 2	30	20	Bob	5	1	5
Run 3	30	22	Bob	6	2	6
Run 4	30	30	Alice	7	1	7
Run 5	30	18	Bob	6	3	6
Run 1	50	28	Bob	7	2	7
Run 2	50	22	Bob	8	1	8
Run 3	50	38	Bob	7	1	7
Run 4	50	50	Alice	8	1	8
Run 5	50	26	Bob	6	2	6

As shown in Table 6.8, as the size of the graph increases (i.e., as the number of vertices grows), Alice's success rate in completing the game without violating the majority rule decreases. Specifically, in graph structures with 30 and 50 vertices, Alice was able to win only once in each case, while in all other instances, Bob created an early majority violation and ended the game prematurely. This suggests that while Alice's strategy is effective in smaller graphs, its performance diminishes in larger instances due to increasing structural complexity, higher vertex connectivity, and reduced control over critical regions.

The analysis also shows that in games where Alice wins, the maximum number of moves is always reached (i.e., all vertices are colored), and a greater number of colors is used. In contrast, games won by Bob are completed in fewer moves and often involve reusing existing colors. This confirms that Bob's goal is not to expand the coloring but simply to create a violation. Moreover, it is observed that in Alice's winning games, the number of colors used increases with the graph size. This suggests that color diversity provides strategic flexibility and plays a critical role in managing challenging situations. These experimental findings demonstrate that the difficulty level of the Majority Coloring Game is directly influenced by the size and structure of the graph. Table 6.9 presents the average values obtained from the experiment.

Table 6.9. Average game metrics.

Vertex Count	AVG Total Moves	AVG Total Colors	Wins
10	8.4	2.4	Alice: 2, Bob: 3
16	14.0	3.6	Alice: 2, Bob: 3
30	21.2	5.8	Alice: 1, Bob: 4
50	32.8	7.2	Alice: 1, Bob: 4

Table 6.10 presents the results of Alice's odd-degree reduction strategy and the Stackelberg strategy, where Alice and Bob alternate as leader and follower on graphs with varying vertex counts. As shown, the average number of moves and colors used

rises with graph size. Alice's odd-degree reduction strategy increases the number of moves Bob needs to win, making the game more difficult for him. Therefore, to challenge Alice more, it would be reasonable for Bob to follow the Stackelberg strategy, acting more aggressively and taking the leader role. Moreover, when the interface is designed, it is expected that if the human player (Alice) consciously applies the odd-degree reduction strategy, they can implement a more effective defense than in the Stackelberg scenario.

Table 6.10. Comparison on two strategies.

Vertex Count	Strategy	Total Games	AVG Moves	AVG Colors
10	Odd-Reduction	5	8.4	2.4
10	Stackelberg	5	4.8	2.4
16	Odd-Reduction	5	14.0	3.6
16	Stackelberg	5	8.0	2.2
30	Odd-Reduction	5	21.2	5.8
30	Stackelberg	5	12.0	3.4
50	Odd-Reduction	5	32.8	7.2
50	Stackelberg	5	20.8	4.6

Although the Odd Degree Reduction strategy improves Alice's performance in simulations, it was not incorporated into the interactive game, where the human player is expected to make real-time decisions as Alice. Still, it was included in the experimental analysis, as managing odd-degree vertices is crucial to understanding structural difficulty and can directly influence game outcomes. Analyzing this strategy under aggressive Bob behavior helps evaluate how difficulty changes and when Alice may gain an advantage.

6.4. Different Graph Families

This section presents a comparative analysis of how different graph structures influence the difficulty of the Majority Coloring Game, under various strategies applied by Alice and Bob. The four graph types under consideration (Cycle, Star, Wheel, and Petersen) are analyzed using the Greedy Strategy. Each graph type was tested with an equal number of simulation runs across vertex sizes (10–40), ensuring fairness in difficulty comparison. While Cycle, Star, and Wheel graphs were tested across multiple vertex counts (10–40), the Petersen graph is a unique 10-node structure with no scalable variants. Therefore, only 5 runs on the same graphs per strategy were conducted for the Petersen graph, as opposed to 20 runs for other graph types. The results are presented in Table 6.11.

In the Greedy strategy, Alice always wins the game. Thus, difficulty can be interpreted in terms of the number of colors required to successfully complete the coloring without violating the majority rule. A higher number of colors implies more defensive actions by Alice and a structurally more challenging graph.

Cycle graphs, where each vertex has exactly two neighbors, exhibited a stable color usage across all sizes tested. The average number of colors used remained around 2.8–3.0 for 10, 20, 30, and 40 vertices. This stability results from the regular, repeating structure of cycle graphs, allowing for frequent reuse of the same colors.

Although Cycle and Wheel graphs show similar average color usage, especially for 30 and 40 vertices, Wheel graphs are structurally more demanding due to the central node that increases connectivity and the risk of majority rule violations. This hub places greater color pressure on Alice, as mistakes or delays around it can more easily lead to failure. Thus, despite comparable color counts, the Wheel's denser and centralized structure makes it moderately more difficult in the Majority Coloring Game.

Table 6.11. Graph families and average colors used.

Graph Type	Vertex Count	AVG Colors Used
Cycle	10	3.0
Cycle	20	2.8
Cycle	30	3.0
Cycle	40	3.0
Star	10	2.0
Star	20	2.0
Star	30	2.0
Star	40	2.0
Wheel	10	3.0
Wheel	20	3.0
Wheel	30	3.8
Wheel	40	3.8
Petersen	10	5.7

In star graphs, all vertices are connected only to a central hub vertex. Simulations have shown that even in a 10-vertex star, on average only two colors are needed, reflecting the extremely low likelihood of majority rule violations under the Greedy strategy. This makes the Star graph the easiest type to color successfully in this context. However, when the number of leaves is odd, Bob can exploit the structure—which consists entirely of odd-degree vertices—to force a violation in more challenging versions. Since Bob does not act aggressively in the Greedy version, the Star graph can always be colored without violating the majority rule, regardless of its size. For this reason, it is strategically more reasonable to include the Star graph only at the easy difficulty level without any color restriction.

The Petersen graph, a highly connected and symmetric graph with 10 vertices, showed higher color usage than the other types tested. Even though it is small in size, its structural complexity that arises from the number of odd-degree vertices makes it difficult to reuse colors effectively. This places the Petersen graph among the more challenging types despite its limited number of vertices.

Based on average color usage, a difficulty ranking while applying Greedy Strategy can be established as:

$$\text{Star} < \text{Cycle} < \text{Wheel} < \text{Petersen}.$$

Graphs with higher connectivity and centralization tend to impose greater constraints, requiring more colors to satisfy the majority rule. These findings highlight that the structural properties of the graph significantly impact the complexity of the Majority Coloring Game.

As star graphs, graphs such as Cycle, Wheel, and Petersen, which have closed structures or contain odd-degree vertices, are only meaningful to include in the game's easy (greedy) version when there is no color restriction. In this scenario, "no color restriction" means that both Alice and Bob can introduce as many new colors as they wish, and Bob can also reuse colors previously used by Alice in his moves. This provides Bob with a critical advantage when playing aggressively: the color distribution that Alice creates for defensive purposes can be exploited by Bob to form same-colored neighborhood chains, thereby violating the majority rule.

Especially in cycle graphs, due to chain-like neighborhood connections, it is quite likely that Bob will violate the majority rule by the fourth move, regardless of the graph's size. An example of this is shown in the Figure 6.1: in a game where Bob plays using the Stackelberg strategy, he is able to win on the fourth move because he has the freedom to reuse colors that Alice has already employed. In this scenario, Alice first colors vertex 1 blue, then Bob colors vertex 0 blue as well. Next, Alice colors vertex 2 orange, but finally Bob colors vertex 9 blue, violating the majority rule.

In a wheel graph, the risk is further amplified by the presence of a central odd-degree vertex in addition to the cycle structure; in the Petersen graph, the 3-regular structure and high symmetry frequently provide Bob with opportunities to form same-colored pairs, putting Alice under significant defensive pressure. While having all vertices with odd degrees is already sufficiently challenging for Alice, Bob's aggressive Stackelberg strategy makes it nearly impossible for her to win.

Therefore, in scenarios where there are no restrictions on the number or types of colors that Alice and Bob can use, these types of graphs should only be included in the game's easy version. This condition still allows Alice to win if Bob plays a more random strategy without systematically trying to trap her. Otherwise, it becomes impossible for Alice to win in Wheel, Cycle, and Petersen graphs under aggressive adversarial play.

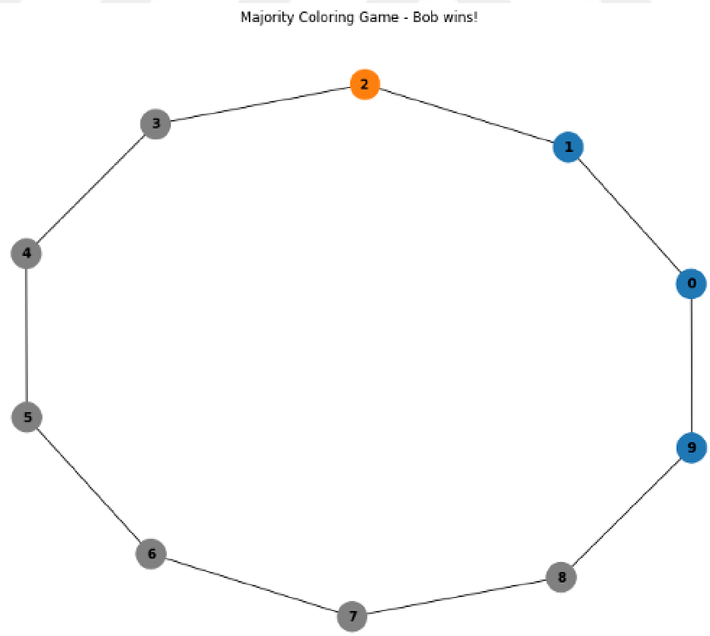


Figure 6.1. Example of a game ending on cycle graph.

Since the distinctive structural properties of the Petersen graph such as its vertex-transitivity and the fact that it consists entirely of odd-degree vertices were observed to significantly influence the dynamics of the game, it is meaningful to extend the analysis to other vertex-transitive graphs structurally similar yet possess important differences.

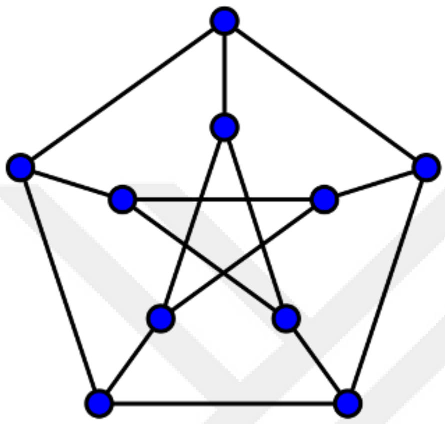
Vertex-transitive graphs, where all vertices are structurally indistinguishable and any vertex can be mapped to any other through an automorphism, are particularly valuable in strategy studies because they minimize local topological bias and emphasize the pure effects of the players' strategies [13]. In this context, the Cubical, Desargues, and Heawood graphs were selected to test the Greedy strategy as well. These graphs, like the Petersen graph, are 3-regular and composed entirely of odd-degree vertices, but they differ in size, girth, and cycle structure, which can affect how easily Bob can create chain-like majority rule violations under aggressive play.

The concept of girth is particularly important. The girth of a graph is defined as the length of its shortest cycle, and for graphs that contain no cycles it is considered infinite [14]. Theoretically, a graph with a smaller girth contains shorter, tightly connected cycles, which can make it easier for Bob to form same-colored neighborhood chains and quickly violate the majority rule.

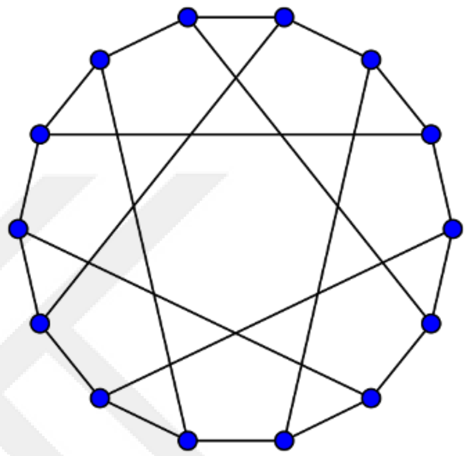
In contrast, graphs with a larger girth contain longer cycles, making it theoretically harder for Bob to create local chain-like risks. However, in practice, this can introduce a different kind of defensive challenge: when a graph is vertex-transitive and highly symmetric, Bob can replicate the same violation risk across multiple regions of the graph by exploiting repeating cycle patterns. This forces Alice to defend multiple cycles simultaneously by diversifying colors more extensively, which increases the total number of colors used.

For example, a small graph like the Cubical graph, with its short girth, makes defense relatively more manageable for Alice, whereas larger graphs such as Heawood and Desargues, with longer girth values, distribute the risk across a wider area, ultimately raising Alice's defensive cost. These findings indicate that even when Bob does not play fully aggressively, structural parameters such as girth together with cycle length, graph symmetry, and size directly influence Alice's defensive strategy and the overall cost of color usage.

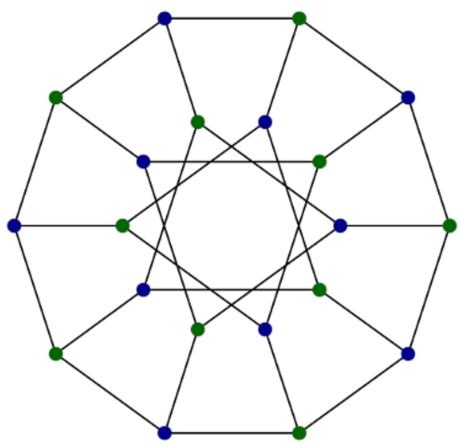
The reason for including this specific subset is to observe how variations in these structural parameters impact Alice's ability to defend against Bob's aggressive moves, especially in a setting with no restrictions on the number or reuse of colors. Below are the figures of the vertex-transitive graphs used in these experiments.



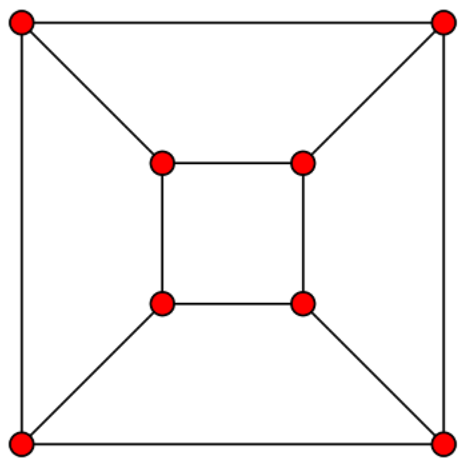
(a) Petersen graph.



(b) Heawood graph.



(c) Desargues graph.



(d) Cubical graph.

Figure 6.2. 3-regular graphs.

Table 6.12. Game difficulty indicators by graph type and strategy.

Graph	**Strategy**	**Vertex Count**	**Girth**	**AVG Colors**	**AVG Moves**
Cubical	Greedy	8	4	4.0	8.0
Desargues	Greedy	20	6	6.1	20.0
Heawood	Greedy	14	6	7.0	14.0

When using the Greedy strategy, Alice always completes the coloring without opposition. In this case, the main indicator of difficulty is how many colors she needs. For each graph type, 10 runs were performed and the averages of the number of colors used were calculated.

In the Cubical graph (3-cube), which has eight vertices and a girth of four, the average number of colors used across ten runs was 4.0. This relatively low value suggests that when Bob plays with only a simple sabotage approach rather than a fully aggressive Stackelberg strategy, Alice can manage to defend the majority rule with minimal color usage. The small size of the graph and its short, square-like cycles allow Alice to break potential violation chains efficiently by diversifying colors only when necessary. Consequently, the Cubical graph illustrates that in smaller, highly symmetric, vertex-transitive structures with short girth, the defensive burden remains relatively light under a nonadversarial scenario.

For the Heawood graph, which has 14 vertices and is 3-regular, vertex-transitive, bipartite, and has a girth of six, the average number of colors used was notably higher at 7.0. Although the longer cycle lengths theoretically give Alice more space to distribute colors defensively, the larger size of the Heawood graph and the extensive symmetry create multiple pathways for Bob to attempt simple sabotage. This means that even without an overly aggressive strategy, Alice must continuously manage potential violation points spread across the graph, resulting in an increased number of colors used compared to the Petersen and Cubical graphs. This indicates that a longer girth alone does not guarantee an easy defense when the graph remains highly symmetric and entirely composed of odd-degree vertices.

The Desargues graph, which is the largest among the tested examples with 20 vertices, is 3-regular, vertex-transitive, and has a girth of six, showed an average color usage of 6.1. Similarly to Heawood, the larger size and longer cycles create a more distributed risk landscape for Alice. Bob's random sabotage strategy can still initiate multiple localized risks, forcing Alice to extend her coloring to prevent majority rule violations throughout the graph. The results suggest that although the larger girth provides slightly buffer for defense, the expanded graph size and structural symmetry increase the overall defensive burden, requiring more colors on average than in the Cubical graph.

6.5. Color Constraints on Game Difficulty

In the Majority Coloring Game, one of the key elements influencing the game's difficulty is the number of colors available to Alice and Bob. In these scenarios, each player is assigned a separate, non-overlapping color set, meaning that neither can use the other's colors. While previous sections have shown that structural properties such as the proportion of odd-degree vertices or the underlying graph topology significantly impact the outcome, this section investigates how restricting the number of usable colors adds further complexity for the player attempting to satisfy the majority rule, namely Alice.

In this experiment, the Stackelberg-based Majority Coloring Game under varying color constraints is simulated, where each player was allowed to use 1, 2, 3, or 4 colors. The goal was to observe how the restriction or expansion of the color set affects Alice's ability to defend against Bob's attempts to violate the majority rule. For each setting, 20 independent runs were conducted using Erdős–Rényi random graphs with 10 vertices and minimum degree constraints to ensure well-formed instances. The results are summarized in Table 6.13.

Table 6.13. Effect of color availability on Alice's success and game length.

Version	Colors per Player	Alice Wins	Bob Wins
1	1	7	13
2	2	9	11
3	3	10	10
4	4	12	8

As the number of available colors increases, a consistent pattern emerges: Alice's success rate improves notably, increasing from 35% under the most restrictive condition (one color) to 60% when she is granted access to four distinct colors. This outcome aligns with theoretical expectations, as a broader color set allows Alice greater flexibility in maintaining the majority rule, particularly by reducing the risk of creating vulnerable configurations and enabling more effective defense of structurally critical nodes that Bob may exploit.

Figures 6.3 and 6.4 illustrate end-game scenarios in a Stackelberg-dominated game where Alice and Bob were assigned 1 and 2 colors, respectively.

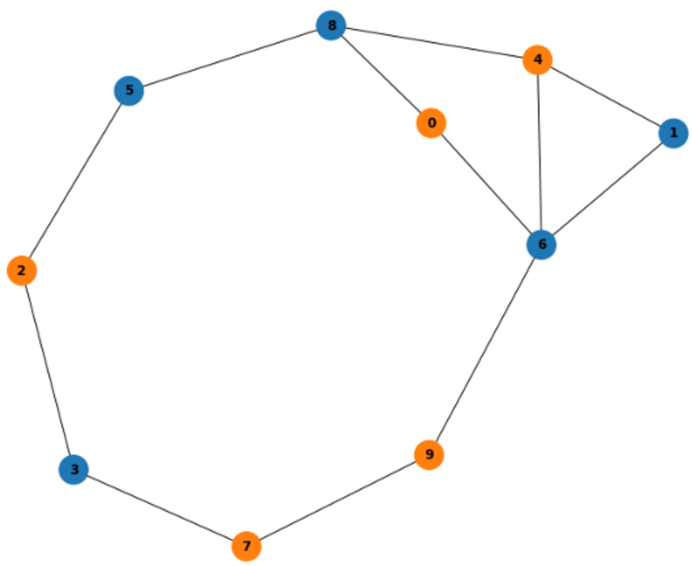


Figure 6.3. Stackelberg game with 1 color.

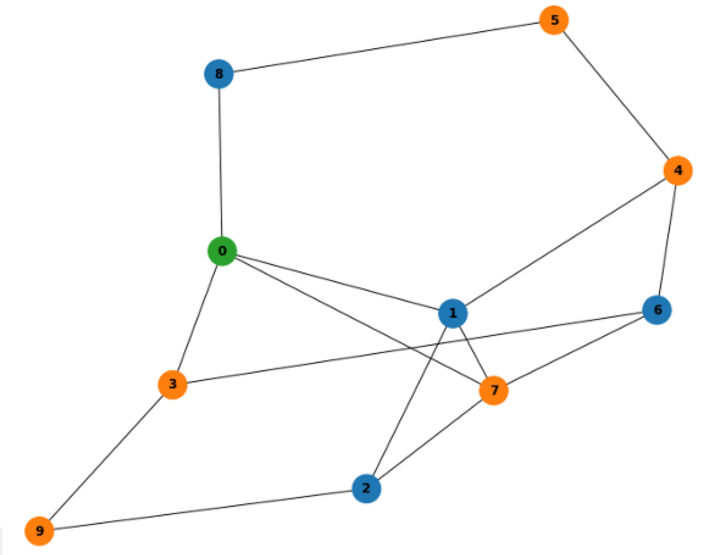


Figure 6.4. Stackelberg game with 2 colors.

As previously introduced, the Greedy version of the Majority Coloring Game provides a simplified setting without full predictive or adversarial planning by either player. In this version, Alice follows a defensive heuristic that prioritizes reusing existing colors when possible and adds new ones only if necessary to avoid violations. Meanwhile, Bob applies a probabilistic sabotage approach: most of the time he tries to choose colors that could lead to future majority rule violations by reinforcing risky neighbor patterns, but occasionally he makes random moves without actively planning sabotage.

The Table 6.14 summarizes the outcomes of 20 independent runs per setting, conducted on 10-node Erdős–Rényi graphs, under varying color constraints ranging from 1 to 4 colors.

Table 6.14. Alice and Bob win rates in the greedy version.

Version	Colors per Player	Alice Wins	Bob Wins
1	1	4	16
2	2	8	12
3	3	20	0
4	4	20	0

When only a single color is available, Alice loses the vast majority of games, with the average game ending in fewer than four moves. This outcome illustrates the limiting effect of strict color constraints: without flexibility in color selection, Alice frequently exhausts her valid move options early, either leading to a violation of the majority rule or being unable to proceed further. Allowing a second color appears to improve Alice's ability to continue the game as seen in Table 6.14. In scenarios where three or more colors are available, Alice succeeded in all trials. These results indicate that, in the absence of adversarial strategies, increasing the number of available colors to three may be sufficient for Alice to maintain control throughout the game. Moreover, when designing the simplest version of the game, it would be much more reasonable to provide three or more colors to Alice (the human player), especially in cases where Alice has not yet mastered the game or developed an effective strategy. The figures 6.5 and 6.6 illustrate end-game scenarios in a Greedy version of the game where Alice and Bob were assigned 1 and 2 colors, respectively.

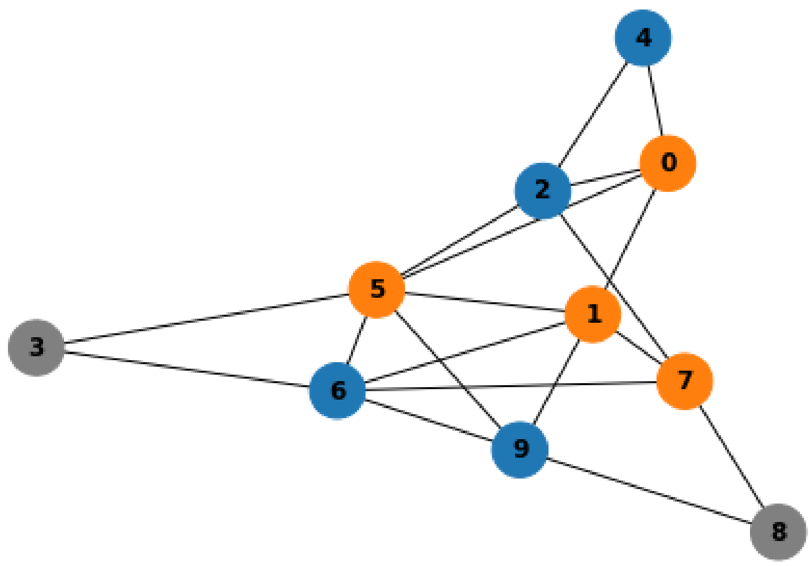


Figure 6.5. Greedy strategy with 1 color.

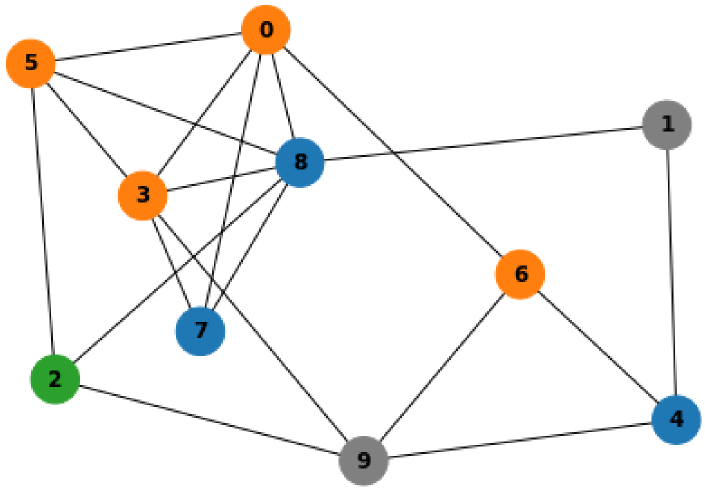


Figure 6.6. Greedy strategy with 2 colors.

A comparison of the Stackelberg and Greedy strategies shows that game difficulty and Alice's performance are closely tied to both the number of available colors and the presence of adversarial pressure. In low-color scenarios (1–2 colors), Bob's tendency to adopt an aggressive approach and make strategic moves is much lower in the Greedy version compared to the Stackelberg version. Despite this, Alice may appear to perform better under the Stackelberg strategy in these constrained scenarios. The main reason for this is that in the Stackelberg approach, Alice can sometimes assume the leader role and develop defensive strategies while Bob is forced to act as a follower responding to her moves. Additionally, the fact that Bob is restricted from using Alice's colors limits his ability to make aggressive plays, which provides a significant advantage to Alice when there are only a few color options. In contrast, the Greedy version lacks a strong sabotage strategy for Bob and does not equip Alice with effective defensive mechanisms either, which can unexpectedly restrict her move flexibility under tight color constraints and make her more vulnerable. However, when more colors are available (3–4 colors), this pattern reverses: Alice wins all games in the Greedy version, while Bob still secures some wins in Stackelberg due to his more targeted aggression. In the Stackelberg scenario, the increased strategic depth and Bob's more disruptive moves can outweigh Alice's defensive advantages, even in high-color settings. These outcomes clearly show that strategic depth and resource constraints together shape the game's difficulty. Tests on graphs of different sizes confirmed the same trend: under tight color limits, the

Stackelberg strategy helps Alice survive longer and win more often, while the Greedy version is more favorable when there is enough color flexibility. Overall, this consistency confirms that the balance between color availability and player strategy fundamentally determines how hard the Majority Coloring Game will be.

Based on these findings, the adjustment of color constraints should be carefully planned to balance the difficulty levels in the game's interface design. In the simple version that uses the Greedy strategy, Bob's capacity for aggressive sabotage is low, and the human player (acting as Alice) is unlikely to have fully developed strategic reflexes yet. If color flexibility is too restricted under these conditions, the player may quickly run out of safe moves and lose the game too easily. Therefore, in the Greedy (easy) mode, giving both Alice and Bob 3–4 colors help preserve the player's decision space and makes the game more learnable and forgiving. In contrast, in the hard version using the Stackelberg strategy, Bob can act as a proactive leader who keeps Alice under constant pressure; giving Alice too many color options in this mode would reduce the intended challenge. Limiting her maneuverability here allows Bob's strategic threats to have greater impact. Thus, imposing a 1–2 color limit in the Stackelberg (hard) mode creates a more meaningful challenge for the player. A color constraint balance designed this way enables the game to offer a smooth learning curve while gradually increasing in difficulty.

6.5.1. Color Constraints in Graph Families

Due to their structural properties, it was previously discussed that in the Stackelberg strategy where Bob plays aggressively and there are no color constraints (meaning Bob can use any color, including those already used by Alice) it is extremely difficult for Alice to win on graphs such as Cycle and Wheel. To further examine this, experiments were conducted by assigning both Alice and Bob 1, 2, 3, and 4 colors each, and running five independent trials for each graph type. The results of these tests are presented in Table 6.15.

Table 6.15. Cycle and wheel graph results.

Graph	n	Alice Colors	Bob Colors	Alice Wins	Bob Wins	AVG Moves
Cycle	10	1	1	2	3	9.6
Cycle	10	2	2	3	2	8.4
Cycle	10	3	3	3	2	8.4
Cycle	10	4	4	3	2	8.4
Cycle	30	1	1	0	5	10.2
Cycle	30	2	2	2	3	15.6
Cycle	30	3	3	3	2	20.4
Cycle	30	4	4	3	2	20.4
Cycle	40	1	1	0	5	12.6
Cycle	40	2	2	2	3	19.6
Cycle	40	3	3	3	2	26.4
Cycle	40	4	4	3	2	26.4
Wheel	10	1	1	0	5	6.0
Wheel	10	2	2	1	4	6.8
Wheel	10	3	3	2	3	7.6
Wheel	10	4	4	2	3	7.6
Wheel	30	1	1	0	5	6.0
Wheel	30	2	2	1	4	10.8
Wheel	30	3	3	2	3	15.6
Wheel	30	4	4	2	3	15.6
Wheel	40	1	1	0	5	6.0
Wheel	40	2	2	1	4	12.8
Wheel	40	3	3	2	3	19.6
Wheel	40	4	4	2	3	19.6

The experimental results clearly demonstrate that, in the Stackelberg Majority Coloring Game with an aggressive Bob, Cycle and Wheel graphs create structurally challenging conditions for Alice, especially as graph size increases. The closed ring structure of the Cycle graph allows Bob to build same-colored neighborhood chains that can violate the majority rule if Alice cannot diversify her coloring in time. In contrast, Wheel graphs add an additional layer of difficulty due to the presence of a central odd-degree hub, which acts as a structural weak point by making majority violations even easier to trigger.

When the number of colors available to both players increase from one to four, Alice's success rate does improve moderately. More colors enable her to spread risk and block Bob's threats more effectively. However, this additional flexibility is not sufficient to guarantee her a win in larger graphs, as structural risks persist. The data shows that, particularly in Cycle graphs, the average number of moves rises significantly with more colors and larger graph sizes, which indicates that Alice is able to prolong the game by defending longer before eventually losing. This highlights how increased color availability can make Bob work harder to force a violation.

A notable insight comes from comparing Bob's performance on both graph types under similar conditions. The results consistently show that, for the same vertex count and color constraints, Bob typically wins in fewer moves on Wheel graphs than on Cycle graphs. This confirms that the odd-degree center node in Wheel graphs makes it structurally easier for Bob to create majority violations quickly, reducing the number of moves needed to win. In contrast, the purely closed structure of Cycle graphs allows Alice to stretch the risk across a wider area, forcing Bob to expend more effort to achieve the same outcome.

Taken together, these findings confirm that structural features such as closed cycles and central odd-degree nodes fundamentally shape the difficulty of the game. While increasing the number of available colors moderately improves Alice's chances and prolongs the game, it does not fully overcome the inherent risks of these graph types. Interestingly, the fact that Bob cannot reuse Alice's colors generally works in Alice's favor when she has multiple colors to choose from, since it limits Bob's ability to replicate same-colored neighborhood chains. However, when both players are restricted to only one color each, this constraint can backfire: Alice can become trapped with no valid moves and be forced to violate the majority rule herself. An example of this situation is shown in Figure 6.7. In a cycle graph with 16 vertices, when Alice and Bob are each assigned only one color, Alice will have no other move left and will be forced to make her last move by coloring vertex 12 or vertex 14 blue; this will cause her to violate the majority rule and lose the game by her own mistake. The observation that Bob sometimes needs more moves to win when Alice loses indicates that greater color flexibility does in fact increase Bob's effort, even if the final outcome remains the same. Overall, the results highlight how the interplay of graph topology, color constraints, and adversarial strategy determines the strategic balance in the game.

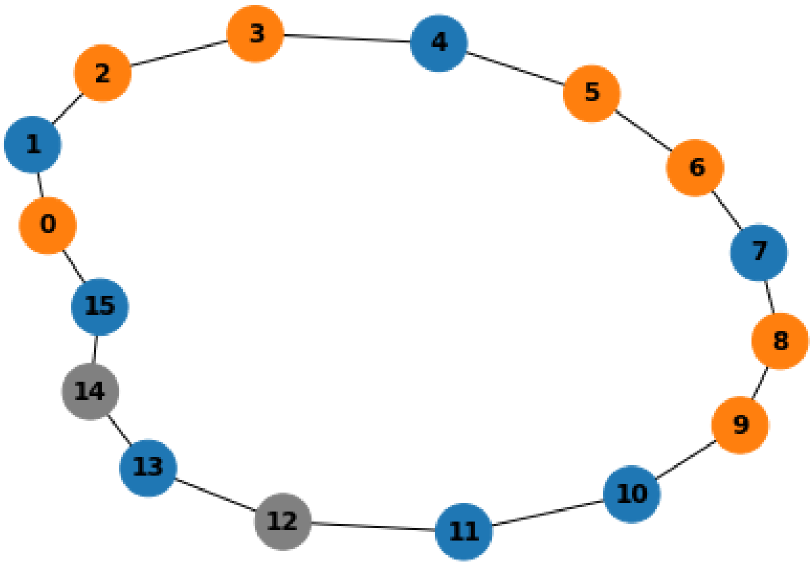


Figure 6.7. Alice's forced self sabotage.

In addition to the Wheel and Cycle graphs, this section also examines how color constraints affect the difficulty level of the Majority Coloring Game in the Petersen, Cubical, Heawood, and Desargues graphs under the Stackelberg strategy. These graphs were shown to be fully colorable under the majority rule in the Greedy strategy scenario, where Bob does not play aggressively and no color constraints are imposed. However, this outcome does not hold under the Stackelberg strategy. Due to the fact that these graphs are 3-regular, Bob's aggressive strategy combined with his ability to reuse the colors chosen by Alice allows him to easily create same-colored neighborhood chains that violate the majority rule. Therefore, this section examines how imposing color constraints by assigning one, two, three, or four colors to each player affects Alice's defensive performance and Bob's chances of winning in these structurally challenging graphs.

Table 6.16. Results for other vertex-transitive graphs.

Graph	n	Alice Colors	Bob Colors	Alice Wins	Bob Wins	AVG Moves
Petersen	10	1	1	0	5	6.0
Petersen	10	2	2	0	5	6.0
Petersen	10	3	3	0	5	6.0
Petersen	10	4	4	0	5	6.0
Cubical	8	1	1	0	5	6.0
Cubical	8	2	2	2	2	6.8
Cubical	8	3	3	2	2	6.8
Cubical	8	4	4	2	2	6.8
Heawood	14	1	1	0	5	6.8
Heawood	14	2	2	0	5	6.8
Heawood	14	3	3	0	5	6.8
Heawood	14	4	4	0	5	6.8
Desargues	18	1	1	0	5	6.8
Desargues	18	2	2	0	5	7.2
Desargues	18	3	3	0	5	7.2
Desargues	18	4	4	0	5	7.2

These combined results clearly demonstrate that vertex-transitive, fully odd-degree, 3-regular graphs such as Petersen, Cubical, Heawood, and Desargues structurally limit Alice's chances of winning under the Stackelberg strategy. The high symmetry of these graphs makes it easier for Bob to strategically build same-colored neighborhood chains that violate the majority rule, and even the color flexibility provided to Alice both in terms of a higher number of colors and Bob's inability to reuse her colors does not fully eliminate this risk. Indeed, when both players were assigned only one color each, Alice was unable to win any runs, and even when the number of colors was increased, she managed to achieve multiple wins only in the Cubical graph. In this context, the Cubical graph stands out as a notable exception. This graph has a low girth, so under normal circumstances, Bob would be expected to quickly turn these short cycles into violation chains in the Stackelberg scenario. However, the fact that the average number of moves remains stable at around 6–6.8, even as more colors are allowed, shows that Alice was able to partially neutralize this short-cycle risk by spreading it over a wider area thanks to the graph's regular and small structure. This demonstrates that the Cubical graph's small size and vertex-transitive nature allow Alice, even in Stackelberg conditions, to limit the short-cycle threat to some extent though not turn it entirely to her advantage. The fact that Alice secured two wins in each of the 2, 3, and 4 color conditions, achieving the highest win count in this class, further supports this point. Interestingly, the same Cubical structure works to Alice's advantage under the Greedy strategy; since Bob does not act aggressively there, he does not actively turn the short cycles into violation chains. As a result, despite its low girth, Alice can achieve full success using only a small number of colors.

By contrast, larger graphs like Heawood and Desargues, which have longer girth values, show that as the number of colors increases, the average number of moves can rise up to 7.2, indicating that while Alice can delay Bob's violation in the color restricted Stackelberg scenario, she still cannot prevent it due to the structural risks posed by these vertex-transitive, fully odd-degree, 3-regular graphs.

Graphs such as Petersen, Cubical, Heawood, Desargues, Wheel, and Cycle are structurally challenging and inherently provide strong advantages to Bob, reducing Alice's chances of winning to nearly zero under the Stackelberg strategy even with strict color constraints. According to the results obtained, it is more appropriate to include these graph structures in the game under the Greedy strategy without color restrictions, or if they are to be used in the harder versions, only the Wheel, Cycle and Cubical graphs can be used in color restricted mode and they should be offered at an intermediate difficulty level with 3–4 color flexibility. This way, even players who have not yet developed advanced strategies still have a realistic chance to win. Otherwise, none of these graph types can be colored without violating the majority rule when Bob plays with his most aggressive Stackelberg strategy.

The Odd-Reduction strategy was excluded from the color-restriction analysis, as its effectiveness is closely tied to Alice's ability to flexibly assign colors in response to structural risks, particularly involving odd-degree vertices. Core mechanisms of the strategy such as reducing uncolored odd neighbors and defending critical nodes often depend on introducing new colors to maintain majority rule compliance. When the number of colors is strictly limited, this flexibility is lost, severely restricting Alice's strategic options and rendering the approach ineffective or unrepresentative. Moreover, for vertex-transitive and 3-regular graphs, the Odd-Reduction algorithm is inherently impractical because all vertices have the same degree and are odd, leaving no structural variation for Alice to exploit. As such, including Odd-Reduction in this context would not produce meaningful or interpretable results under constrained conditions.

6.6. Majority Chromatic Number and Majority Coloring Number

The thesis also aims to investigate the relationship between the game outcomes and the theoretical majority chromatic number $\mu_{\mathrm{g}}(G)$, a parameter which indicates the minimum number of colors required to color a graph without violating the majority rule.

In order to investigate the relationship between the majority chromatic number ($\mu_{\mathrm{g}}(G)$) and the actual gameplay dynamics in the Majority Coloring Game, several structured graphs with known theoretical upper bounds were selected. These include cycle, star, complete, wheel, and cubical graphs. The Greedy strategy, where Alice consistently wins without any color restriction, was employed to ensure that the entire graph is colored in accordance with the majority rule. This was crucial for reliably investigating the majority chromatic number in a controlled, violation-free context. The results demonstrated that in most cases, the number of colors used during gameplay remained within the theoretical upper bound, validating the experimental setup.

To provide context for the experimental results, theoretical upper bounds of the majority chromatic number $\mu_{\mathrm{g}}(G)$ can be summarized for several well-known graph families. These values are drawn from existing literature and serve as benchmarks for evaluating the performance of experimental strategies. For specific graph types, some known bounds are presented in Table 6.17.

Table 6.17. Known theoretical bounds for majority chromatic number.

Graph Type	**Known Theoretical Bound** $\left(\mu_{\mathrm{g}}(\mathrm{G})\right)$
Cycle Graph ($n \geq 4$)	≤ 3
Star Graph	≤ 2
Complete Graph K_n	$\leq n$
Wheel Graph	$\leq$ 4–5 (typically $\Delta + 1$)
Cubical Graph (3-regular)	≤ 4

Table 6.18 presents an analysis of the majority chromatic number for various graphs under the condition that the entire graph is colored in compliance with the majority rule in the greedy strategy. To enable a more robust analysis, multiple runs were conducted for each graph and the findings were evaluated accordingly. To test the stability of the Greedy strategy in which Alice tries to minimize the colors used, the simulation was repeated five times for each graph. The average number of colors used was compared to the known theoretical bounds. All of the graph types consistently adhered to the bounds.

Table 6.18. Results for $\mu_g(G)$.

Graph	**Known** $\boldsymbol{\mu_g(G)}$ **(UB)**	**AVG Colors**	**Bound Match**
Cycle Graph ($n = 6$)	3	3.0	5/5
Star Graph ($n = 6$)	2	2.0	5/5
Complete Graph ($n = 4$)	4	4.0	5/5
Wheel Graph ($n = 6$)	4	4.0	5/5
Cubical Graph	4	4.0	5/5

For a more comprehensive analysis, Bob's strategy was modified by lowering the random move probability used in the Greedy strategy from 20% to 10%, making Bob play more aggressively and in a more targeted way. Each graph was simulated five times under this more aggressive setting. Although this approach increased the number of colors used in some cases (notably in cycle, wheel, and cubical graphs), Alice was still able to win all the games.

Table 6.19. Modified version results for $\mu_g(G)$.

Graph	**Known $\mu_g(G)$ (UB)**	**AVG Colors**	**Alice Wins**
Cycle Graph ($n = 6$)	3	4.0	5/5
Star Graph ($n = 6$)	2	2.0	5/5
Complete Graph ($n = 4$)	4	4.0	5/5
Wheel Graph ($n = 6$)	4	5.0	5/5
Cubical Graph	4	5.0	5/5

The results show that the theoretical upper bounds for the majority chromatic number hold in most cases, but in some graphs Alice's defensive strategy can exceed these bounds in practice. This indicates that structural features and Bob's aggressive moves can push the required number of colors beyond the expected limit. In all tested graph types, Alice was able to complete the coloring without violating the majority rule under both normal and more aggressive conditions. These findings confirm that as the game becomes more challenging and Bob's strategy grows more aggressive, keeping the number of colors within the theoretical bounds becomes increasingly difficult.

After analyzing $\mu_g(G)$, the focus shifts to the majority game coloring number $\mathrm{col}_g(G)$, which reflects the influence of adversarial vertex ordering. The coloring number $\mathrm{col}(G)$ of a graph is defined as the minimum integer $k + 1$ such that there exists a linear ordering of the vertices where no vertex has more than k neighbors earlier in the order. In adversarial settings like the Majority Coloring Game, Bob selects this order, potentially maximizing Alice's back degree. Therefore, the empirical majority game coloring number, $\mathrm{col}_g(G)$, can provide an estimate of the minimum number of colors Alice may need under the worst-case vertex orderings, taking into account both the backward neighbor constraints and the majority rule. However, since the majority rule allows Alice to share colors with some neighbors as long as it does not create a local majority, the backward neighbor count does not always directly determine the exact number of required colors.

Table 6.20. Results for $\mathrm{col}_g(G)$.

Graph Type	**Known col(G)**	**Empirical col$_g$(G)**	**Alice Wins**
Cycle ($n = 10$)	3	3	✓
Star ($n = 10$)	2	10	✓
Complete ($n = 6$)	6	6	✓
Wheel ($n = 10$)	4–5	10	✓
Cubical (3-regular)	4	4	✓

As seen in Table 6.20, in the case of the cycle graph with 10 vertices (even-sized), its bipartite structure leads to a classical chromatic number of 2. However, due to local adjacency, its coloring number is 3, meaning no vertex can avoid having up to two backward neighbors in any order. The simulation confirms that the majority game coloring number matches this value. The graph's symmetry enables Alice to maintain manageable back connections, even when Bob dictates the sequence.

By contrast, the star graph shows a sharp discrepancy. While its chromatic and coloring numbers are both 2, the empirical $\mathrm{col}_g(G)$ spikes to 10. This situation arises when Bob delays the central hub node until the final turns while Alice colors the peripheral nodes first, causing the hub's backward degree to reach up to 9. However, according to the theoretical framework, this scenario would be prevented if Alice plays optimally; therefore, this high value can only be observed in practical settings where Alice's control over backward connections is suboptimal. This illustrates how structural asymmetry and unfavorable vertex orderings can significantly increase the game's difficulty depending on the strategy.

In the complete graph K_6, all vertices are connected to one another. Consequently, the chromatic number, coloring number, and empirical majority game coloring number all align at 6. Regardless of the vertex order, the maximum back degree is always 5, and Bob cannot worsen the scenario. This structural saturation nullifies the effect of vertex reordering.

The wheel graph, which integrates a cycle with a central hub, displays similar behavior to the star graph in this setting. Although its classical coloring number is typically around 4–5, the $\mathrm{col}_g(G)$ rises to 10. When the central node is delayed, it accumulates back connections from all cycle vertices, significantly increasing its local back degree. This again highlights how dynamic orderings in a game setting can inflate color demands beyond theoretical expectations.

Finally, the cubical graph, known for its 3-regular and symmetric structure, displays consistent results. Its classical coloring number is 4, which matches the $\mathrm{col}_g(G)$ observed during simulations. Even when Bob controls the ordering, the balanced nature of the graph ensures no vertex suffers from disproportionately high back degree, keeping color requirements stable.

Overall, the results confirm the expected relationship: $\mathrm{col}_g(G) \geq \mathrm{col}(G)$. This inequality arises from the added difficulty introduced by adversarial vertex ordering. While graphs with regular and symmetric structures (like cycle and cubical) maintain their theoretical bounds, those with centralized or asymmetric topologies (like star and wheel) are far more sensitive to vertex order, leading to substantial increases in required colors. This supports the conclusion that $\mathrm{col}_g(G)$ captures more than structural complexity and it also reflects the impact of adversarial play and decision sequencing within the Majority Coloring Game framework.

It should be noted that the empirical $\mathrm{col}_g(G)$ values presented in Table 6.20 were calculated for each graph using an example vertex ordering observed during a Greedy strategy scenario where Alice successfully completed the coloring without violating the majority rule. This empirical approach was chosen because players rarely adopt a perfect marking strategy in a practical game setting. The aim is to illustrate the practical difficulty level under a realistic Greedy scenario where the majority rule holds. This provides insights into how structural asymmetries and ordering sensitivity affect the coloring outcome, even though the results do not represent the theoretical worst-case upper bound.

Table 6.21. Factors affecting difficulty in the majority coloring game.

Factor	**Impact on Difficulty**
Graph size (↑)	Increases
Number of odd-degree vertices (↑)	Increases
Centralization of odd-degree vertices (↑)	Decreases
Color constraint (↑)	Increases
Bob's probability of random moves (↑)	Decreases
Structural complexity (↑)	Increases

The experimental results summarized in table 6.21 demonstrate that the difficulty level of the Majority Coloring Game for Alice is closely linked to both the structural characteristics of the graph and the gameplay constraints. Larger graphs increase the complexity by expanding the decision space and the number of potentially risky configurations. A higher proportion of odd-degree vertices further exacerbates this challenge, as such nodes are more likely to trigger violations of the majority rule. However, when these odd-degree vertices are centralized rather than scattered, the overall difficulty decreases, allowing Alice to apply more focused defensive strategies.

In the version of the game, where there is no color constraint, Alice and Bob are free to use any colors they wish, and Bob can reuse the colors chosen by Alice to more easily build same-colored neighborhood chains. This increases the impact of Bob's aggressive strategy and makes it more challenging for Alice to defend effectively.

In contrast, in the constrained version where Alice and Bob are assigned separate color sets and neither player can use the other's colors, Bob's capacity to make aggressive moves is limited, providing Alice with a potential strategic advantage. However, as the number of available colors decreases, Alice's flexibility also diminishes, making it harder for her to complete the coloring without violating the majority rule. On the other hand, in the unrestricted version, as long as Alice does not attempt to minimize the number of colors used, greater color flexibility helps make the game more manageable for her.

The findings show that under the Greedy strategy without color constraints, graphs such as Cubical, Petersen, Heawood, Desargues, Wheel, and Cycle, which are highly symmetric and fully odd-degree, can be easily colored by Alice in line with the majority rule. However, under the Stackelberg strategy, Bob can exploit cycles and odd-degree nodes more effectively. Therefore, when using these graphs in harder versions of the game, applying a color constraint that prevents Bob from reusing Alice's colors reduces his aggressive potential and gives Alice's defenses a better chance to succeed. In general, as structural complexity and odd-degree proportions rise and color availability drops, the game becomes harder for Alice, while localizing threats or limiting Bob's aggressiveness helps keep it manageable.

7. Interface Design

To experimentally implement the Majority Coloring Game developed in this study, a comprehensive simulation interface was created using the Unity game engine. Unity was chosen due to its strong support for 2D visual rendering, modular component-based architecture, and powerful C# scripting capabilities, which together facilitated the modeling of interactive graph structures, processing of user inputs, and management of AI-controlled player behavior [15].

This interface not only enables seamless user interaction but also provides a controlled experimental platform where theoretical concepts such as majority rule constraints, graph topologies, and strategic decision-making can be effectively tested and observed in practice.

7.1. System Architecture and Gameplay Flow

The system is structured around Unity's GameObject-based architecture, where each visual or functional element consists of components such as *SpriteRenderers* and scripts derived from *MonoBehaviour*. The interface contains two primary layers: a level selection screen where users choose the difficulty, and a gameplay screen where the graph, color palette, additional buttons, and turn-based logic are presented.

The main scripts and their responsibilities are summarized as follows:

- GameController.cs: Central game logic including Bob's behavior and win/loss evaluation.
- NodeGraphGenerator.cs: Generates random graphs based on the difficulty level.
- GraphManager.cs: Generates predefined graph types such as Cycle, Star, Petersen, etc.
- NodeScript.cs: Handles user interactions with graph nodes.
- GameTextDisplayer.cs: Manages the timer and player turn display.

Upon selecting a difficulty level, the system generates either a random or predefined graph and initializes the set of available colors. Colors can be selected through interactive buttons that players can adjust dynamically using the *Reset Colors* option regardless of difficulty. Players act as Alice and color nodes according to their selected colors. After each move, the game checks for violations of the majority rule. If no violations occur and all nodes are successfully colored, Alice wins. Conversely, if any node violates the majority threshold, Bob wins. Gameplay alternates turns between Alice (the user) and Bob (the AI).

At the start of the game, players are given the option to enable the *Color Restricted Version* through a toggle (checkbox). If activated, Alice and Bob each play using separate, limited color sets defined by the selected difficulty level.

The flowchart below illustrates the operational flow of the Majority Coloring Game and demonstrates how user interactions on the frontend integrate with backend logic implemented through various scripts. In this structure, Alice's moves and Bob's simulated responses are processed within backend modules and dynamically reflected in the game interface. This architecture transforms the theoretical game model into an interactive simulation, allowing systematic testing of strategy performance and observing how changing parameters affect gameplay outcomes. Overall, this flowchart in Figure 7.1 aids understanding by clearly presenting core operational steps, data flow between frontend and backend, and the functioning of backend scripts at each stage.

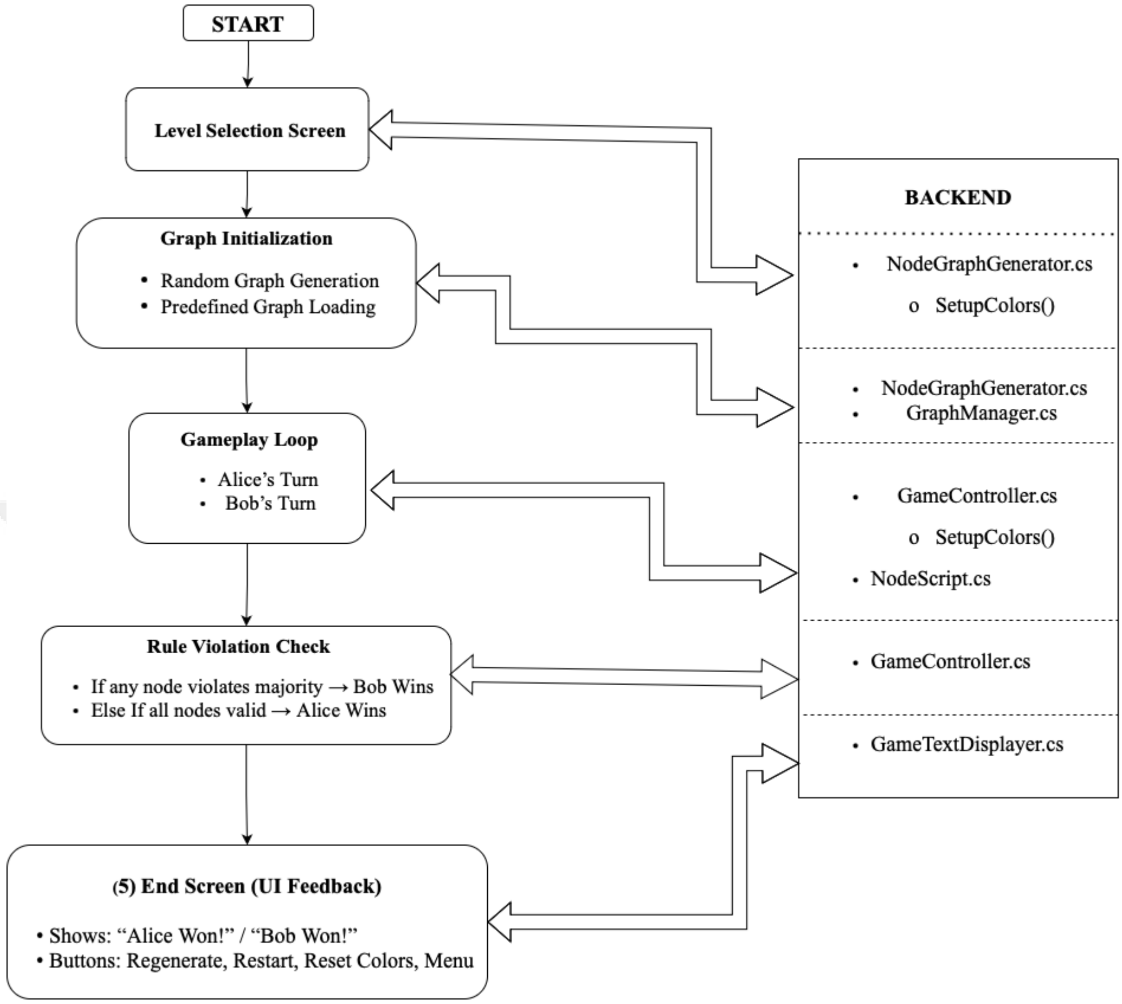


Figure 7.1. The flowchart.

7.2. Visual Interface Components

The interface of the Majority Coloring Game was designed to be both functional and intuitive, presenting key visual components that facilitate real-time interaction between the player and the game system. The user experience is structured around Unity's Canvas system, which organizes all 2D UI elements on the screen. These components are dynamically activated or updated depending on the stage of the game.

(i) Level Selection Panel

Upon launching the game, players are presented with a level selection panel, where they can choose among Easy, Medium, and Hard difficulty levels. This panel contains three interactive buttons, each linked to graph and color configuration functions in *NodeGraphGenerator.cs*. The panel also includes a checkbox option that allows the player to enable the *Color Restricted Mode*, which assigns Alice and Bob separate, limited color sets based on the selected difficulty level as in Figure 7.2. Upon selection, this panel is hidden, and the gameplay panel is displayed as in as seen in Figure 7.3.

Figure 7.2. Level selection menu.

(ii) Gameplay Panel

The gameplay screen displays:

- The graph, rendered with node *GameObjects* connected by *LineRenderer* components.
- A turn indicator text positioned above the graph.
- A color palette, represented by square buttons aligned horizontally below the graph.
- A real-time timer indicating total elapsed time since the start of the game.

(iii) Color Palette Buttons

The color selection panel is created automatically using buttons.

- In the normal version of the game, players can freely choose how many colors they want to use (between 2 and 6) and can change this at any time by clicking the Reset Colors button.

- In the Color Restricted Version, Alice and Bob each get a separate, fixed number of colors depending on the difficulty level.
- Each color button visually represents its color, and when the player clicks a color and then a node, that node is colored with the selected color.

(iv) Graph Interaction and Node Coloring

Each node in the graph is implemented as a clickable GameObject with a *NodeScript.cs* component attached. When clicked:

- If it is Alice's turn and the node is uncolored, it is painted using the selected color.
- The game state is immediately updated, and control passes to the AI (Bob).

This setup creates a highly interactive and visually responsive gameplay loop. As colors are applied to nodes, the visual state of the graph changes, providing immediate feedback to the player. Additionally, the turn-based flow is reinforced by changing textual indicators and by pausing briefly during Bob's moves.

(v) Additional Control Buttons

In addition to the main gameplay mechanics, the interface includes the following auxiliary buttons to enhance user control and experimental flexibility:

- Regenerate: Creates a new randomly generated graph with the same difficulty settings. Useful for testing different graph topologies under identical color constraints.
- Restart: Resets the current graph to its initial uncolored state, allowing the player to replay the same configuration.
- Reset Colors: Generates a new set of colors from the global color pool, with varying number of colors.
- Menu: Returns the user to the level selection screen, allowing switching between difficulty modes.

These buttons provide practical tools for both gameplay iteration and structured experimentation, allowing players to test various strategies and conditions within a single session as seen in Figure 7.3.

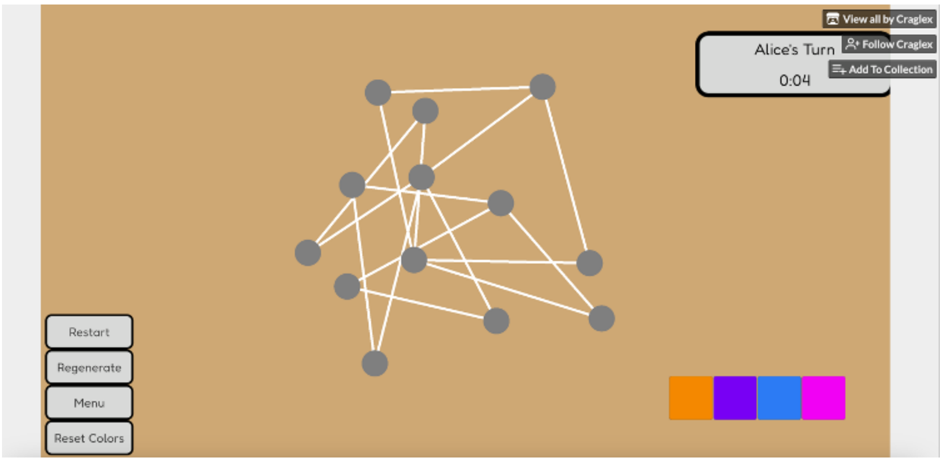


Figure 7.3. Gameplay panel.

(vi) Layout and Visual Consistency

All UI elements are anchored and scaled appropriately to ensure consistent layout across different screen resolutions. The Canvas operates in Screen Space – Overlay mode, which guarantees that UI elements remain fixed and accessible.

7.3. Difficulty Configuration

The game features three difficulty levels *Easy*, *Medium*, and *Hard*, each modifying the number of nodes, odd degree vertex density, configuration, and number of available colors as shown in Table 7.1. Graphs are primarily generated randomly, using spatial placement and proximity-based edge connections. However, each level also includes curated graph types that correspond to theoretical investigations conducted in this study.

Table 7.1. Difficulty configuration.

Level	**Vertices**	**OD Density**	**OD Location**	**Colors**	**Restriction**
Easy	10–15	Low	Centralized	2–6	4
Medium	15–30	Moderate	Moderate Shuffle	2–6	3
Hard	30–50	High	Heavy Shuffle	2–6	2

Importantly, the game provides flexibility in how color constraints are applied. Users are not restricted to a fixed number of colors for each graph. For both randomly generated and predefined graphs, the same structure can be played using different color counts, allowing the player to choose how constrained the game should be. This enables rich experimentation by allowing a random graph to be played with 2, 3, or more colors to observe how coloring flexibility influences game difficulty. In the standard version, the color count is flexible (2–6 colors shared), whereas in the Color Restricted Mode, Alice and Bob each have separate, fixed color sets defined per difficulty level: Easy (4 each), Medium (3 each), Hard (2 each). This level of control not only enhances player agency and replayability, but also supports experimental investigations into how color limitations interact with structural complexity in adversarial coloring scenarios.

Graphs such as Cycle, Petersen, Wheel, Cubical, Heawood, Desargues, and Star have been generated in granular form and are included in the "Easy" difficulty level when no color restriction is applied. However, when Cycle, Cubical and Wheel graphs are added to the higher difficulty level as "Medium", they are only accessible in the version with color constraints; this is because experimental findings suggest that otherwise it would be practically impossible for Alice to win the game.

7.4. Strategic Behavior and Probabilistic Move Mechanism

In this study, Bob's moves during the game are determined by two core algorithmic strategies that operate independently of any human control. These strategies directly parallel the Greedy and Stackelberg approaches developed in this thesis. Depending on the selected difficulty level, Bob's behavior is governed by a probabilistic mechanism that switches between these strategies.

(i) Greedy Strategy (Easy Level): This strategy exactly matches the Greedy approach defined in the thesis. In the Greedy strategy, Bob has two possible modes of behavior:

- *Random:* Bob selects a node randomly and assigns it a color without analyzing any possible outcomes. This passive behavior allows Alice to experiment with her defensive strategies freely.

- *Strategic:* Bob acts more actively by examining neighboring nodes and making a simple local evaluation of which color choice might pose a higher risk for Alice. However, there is no deep optimization here; only a basic local assessment is performed.

This setup covers the possibilities for Bob to play both passively and at a basic active level, giving Alice the opportunity to develop her defense by experiencing different risk environments.

(ii) Stackelberg Strategy (Hard Level): This strategy aligns fully with the Stackelberg scenario defined in the thesis. In this case, Bob normally assumes the leader role by analyzing the structural weaknesses of the graph to force Alice into making risky or constrained moves. Among all possible moves, the one that brings Alice closest to violating the majority rule is selected and executed. However, as a core feature of the Stackelberg dynamic, if Alice (the user) makes a move and takes the leader role, Bob responds to this situation as a follower, making a defensive move in return. This allows the leader–follower role distribution to shift dynamically according to the situation, accurately reflecting the interactive nature of the Stackelberg game.

These two strategies are blended according to the following probability distribution based on the selected difficulty level:

- Easy: 80% Greedy, 20% Stackelberg
- Medium: 30% Greedy, 70% Stackelberg
- Hard: 100% Stackelberg

Through this structure, Bob plays passively or at a simple active level on lower levels, but as the difficulty increases, he fully transitions into the leader role and becomes a strategic, aggressive opponent. If the user makes a strong move to assume the leader position, the Stackelberg dynamic requires Bob to switch to defense and act as a follower. In this way, Alice (the user) can experience different strategic scenarios and develop her decision-making and defensive skills.

7.5. User Interface Feedback and Deployment

The user interface was designed not only to facilitate seamless interaction but also to provide continuous and structured feedback throughout gameplay. It plays a central role in ensuring that players are consistently informed about the current state of the game through dynamic visual elements. These include turn indicators, a real-time timer, and end-of-game result messages, all of which contribute to the clarity of the turn-based structure and assist the player in making timely and informed decisions. Beyond improving usability, the interface functions as a core component of the experimental setup, offering researchers the ability to observe decision-making behavior under different graph structures and constraints. The user interface includes the following core components:

- Turn indicator: a dynamic text field positioned above the graph that displays whose turn it is: "Alice's Turn" when it is the player's move and "Bob's Turn" during the AI phase. This helps maintain clarity in the turn-based structure.
- Timer display: a real-time timer, updated every second, that tracks the total duration of the game session. It provides temporal context, which is particularly useful in analyzing the time efficiency of player strategies or AI responses.
- Result notification: at the end of each game, a message appears indicating the winner ("Alice Won!" or "Bob Won!"), depending on whether the majority rule was preserved or violated. This feedback is triggered immediately upon the detection of a win or loss condition as in Figure 7.4.

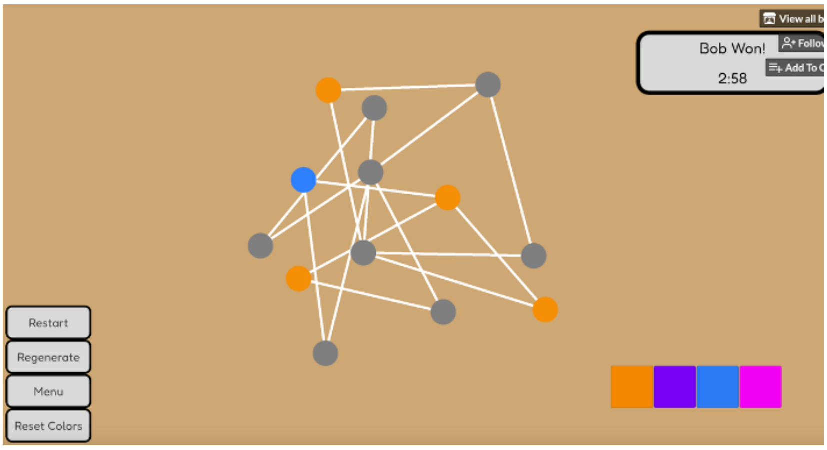


Figure 7.4. End-of-game scenario where Bob is the winner.

All feedback elements are implemented using Unity's UI system (TextMeshProUGUI) and are managed through the *GameTextDisplayer.cs* script, which updates visual elements according to the game state monitored by *GameController.cs*.

Once development was completed, the game was compiled into a WebGL build using Unity's export tools and deployed on the online platform itch.io [16]. This deployment serves multiple purposes: it allows users to access and play the game directly through their web browsers without requiring installation; it enables widespread testing with real-time user interaction; and it offers a publicly available platform for researchers, educators, and students to explore majority coloring behavior under varying graph structures and color constraints. Through this deployment, the simulation becomes a versatile tool that bridges theoretical modeling and interactive experimentation, supporting both educational applications and research-oriented analysis. The WebGL version of the game is publicly accessible at the following URL: https://craglex.itch.io/majority-color-game

8. CONCLUSION

This thesis approaches the concept of the Majority Coloring Game from both theoretical and experimental perspectives, aiming to develop a deeper understanding of its structural dynamics, strategic complexity, and practical implications. The existing literature on majority coloring, including its defined rules, parameter bounds, and theoretical results, forms the basis of this study. Building on this foundation, original strategy models were developed to more realistically reflect the dynamics of the game. Within this scope, a game-theoretic version was formulated in which two players, Alice and Bob, take turns coloring the vertices of a graph under majority constraints.

One important contribution of this study is the development of a structured and interactive game framework that makes it possible to simulate majority coloring dynamics under competitive conditions. In order to capture different heuristic approaches and reactive mechanisms, three original strategies, named Greedy, OddReduction, and Stackelberg, were designed and implemented. These strategies made it possible to evaluate Alice's defensive performance against Bob's attempts to create rule violations under various scenarios. Moreover, different graph types were tested in competitive scenarios, allowing for a systematic comparison between theoretical majority chromatic number upper bounds and practical gameplay dynamics. This revealed how structural properties and player strategies can push color requirements beyond theoretical limits.

The experimental part of the study systematically examines how the difficulty level of the game is affected by the structural features of the graph and the applied constraints. The results show that as the graph size increases, the decision space expands and the number of potentially risky configurations grows, which significantly increases the strategic complexity of the game. An increase in the proportion of odd-degree vertices raises the challenge for Alice, since these nodes are more likely to cause violations of the majority rule. However, when these vertices are grouped centrally rather than scattered, the overall difficulty decreases, allowing Alice to focus her defense more effectively.

In the basic version of the game, both Alice and Bob share the same color pool and can use as many colors as needed. In this case, the absence of a color constraint theoretically increases Alice's chances of coloring all vertices without violating the majority rule, since she can always introduce new colors to avoid risky situations when necessary. Especially when faced with challenging parameters such as graph size or odd-degree vertex configurations, Alice's flexibility in using colors provides a significant advantage. The results show that as the game becomes more complex, the number of colors used by Alice increases. For certain graph types, including vertex-transitive and 3-regular graphs as well as cycle and wheel graphs, Alice was able to complete the coloring without violating the majority rule when applying the Greedy strategy without color restrictions. In contrast, when the same graph types were tested under the more challenging Stackelberg strategy, a color separation constraint became necessary to prevent Bob from using Alice's colors. This restriction limits Bob's ability to create same-colored neighborhood chains that could lead to rule violations and gives Alice a defensive advantage, but this advantage only becomes meaningful if Alice still has enough flexibility in her color choices. This situation applies mainly to the Wheel, Cycle, and Cubical graphs. For highly symmetric and 3-regular graphs such as Petersen, Desargues, and Heawood, the results show that it is not feasible to complete the game in accordance with the majority rule against an aggressive Bob acting as the leader in the Stackelberg strategy. Moreover, even for the Wheel, Cycle, and Cubical graphs, if the number of available colors is severely limited, Alice's defensive options become restricted and Bob's structural advantage once again becomes decisive. For this reason, offering some graph types without color constraints in the Greedy version, or ensuring sufficient color flexibility if a color separation constraint is applied in the Stackelberg scenario, ensures that the game remains strategically challenging yet realistically winnable.

In addition to the theoretical modeling and experimental analysis, an interactive game interface was developed using Unity. This interface supports different difficulty levels, graph types, and parameter configurations, providing a user-friendly platform for exploring the dynamics of the Majority Coloring Game. In the interface, Greedy and Stackelberg strategies are used in varying proportions to create a gradual increase in difficulty. For the easy level, the game uses 80 percent Greedy and 20 percent Stackelberg; the medium level applies 30 percent Greedy and 70 percent Stackelberg; and the hard level is fully based on the Stackelberg strategy. This design allows users to experience different gameplay scenarios depending on strategic variation. Furthermore, the interface has pedagogical value for teaching graph theory and strategic thinking and serves as a practical tool for advanced experimentation.

In conclusion, this thesis combines formal modeling, strategy design, experimental evaluation, and interface development to present a comprehensive study of the Majority Coloring Game. The findings provide valuable insights into how structural features and game parameters affect the game's complexity and offer a solid basis for future work in algorithmic game design, competitive graph coloring, and interactive educational tools. Additionally, the systematic testing of different graph types to relate theoretical majority chromatic number bounds with practical gameplay dynamics stands out as an original contribution of this research.

REFERENCES

1. Lovász, L., "On Decomposition of Graphs", *Studia Scientiarum Mathematicarum Hungarica*, Vol. 1, pp. 237–238, 1966.

2. Sands, B., N. Sauer and R. Woodrow, "On Monochromatic Paths in Edge-Coloured Digraphs", *Journal of Combinatorial Theory, Series B*, Vol. 33, No. 3, pp. 271–275, 1982.

3. Zhu, X., "The Game Coloring Number of Planar Graphs", *Journal of Combinatorial Theory, Series B*, Vol. 75, No. 2, pp. 245–258, 1998.

4. Bosek, B., J. Grytczuk and G. Jakóbczak, "Majority Coloring Game", *Discrete Applied Mathematics*, Vol. 255, pp. 15–20, 2019.

5. Anholcer, M., B. Bosek and J. Grytczuk, "Majority Coloring of Infinite Digraphs", *Acta Mathematica Universitatis Comenianae*, Vol. 88, No. 3, pp. 371–376, 2019.

6. Wang, M. and S. Wu, "A Note on Majority Coloring of Digraphs with Large Indegree", *International Journal of Mathematics Trends and Technology*, Vol. 69, No. 1, pp. 33–37, 2023.

7. Shi, M., W. Xia, M. Xiao and J. Wang, "The Majority Coloring of the Join and Cartesian Product of Some Digraphs", *MATEC Web of Conferences*, Vol. 355, p. 02004, 2022, conference held in Xiamen, China.

8. Destacamento, C. J., A. D. Rodriguez and L. Aquino-Ruivivar, "Game Chromatic Numbers of Graphs: A Review and Some New Results", *Journal of Mathematical Research with Applications*, Vol. 34, No. 2, pp. 176–184, 2014.

9. Bartnicki, T. and J. Grytczuk, "The Game Chromatic Index of Graphs", *Graphs and Combinatorics*, Vol. 24, No. 1, pp. 1–4, 2008.

10. Blitzstein, J. and P. Diaconis, “A Sequential Importance Sampling Algorithm for Generating Random Graphs with Prescribed Degrees”, *Internet Mathematics*, Vol. 6, p. Article 557277, 2010.

11. Jiang, W., H. Han, M. He and W. Gu, “When Game Theory Meets Satellite Communication Networks: A Survey”, *Computer Communications*, Vol. 217, pp. 208–229, 2024.

12. Başar, T. and G. J. Olsder, *Dynamic Noncooperative Game Theory*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2nd edn., 1999.

13. Lorimer, P., “Vertex-Transitive Graphs: Symmetric Graphs of Prime Valency”, *Journal of Graph Theory*, Vol. 8, No. 1, pp. 55–68, 1984.

14. West, D. B., *Introduction to Graph Theory*, Prentice Hall, Upper Saddle River, NJ, 2nd edn., 2001.

15. Unity Technologies, “Unity (Version 2022.3 LTS) [Computer Software]”, `https://unity.com`, 2025, accessed: August 2025.

16. itch.io, “Majority Coloring Game [Web-based Simulation]”, `https://craglex.itch.io/majority-color-game`, 2025, accessed: August 2025.