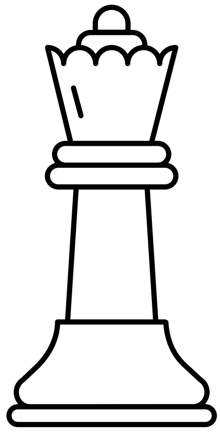
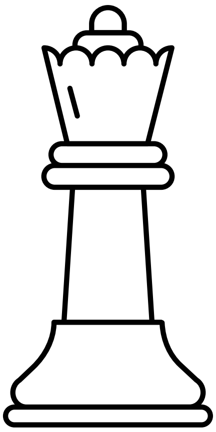
N - Q U E E N



GROUP MEMBERS

Alishbah Rizwan (CS-21044)
Narmeen Shebaz (CS-21047)
Mustafa Mehdi (CS-21061)
Wali Shajeeh (CS-21061)



T.E. (CIS) CEP REPORT

DEPARTMENT OF COMPUTER AND INFORMATION SYSTEMS ENGINEERING

ABSTRACT

This project introduces an innovative approach to solving the classic N-Queens problem through the development of an AI-powered visualizer. The N-Queens problem, a well-known combinatorial puzzle, involves placing N chess queens on an N×N chessboard in such a way that no two queens threaten each other. Our solution leverages artificial intelligence techniques to efficiently explore and visualize potential solutions, providing users with a dynamic and interactive tool for understanding the complexities of the N-Queens problem.

CONTENTS

CHAPTER 1

P R O B L E M   D E S C R I P T I O N

1.1 Historical Significance

The N-Queens problem holds historical significance as one of the classical problems in combinatorial mathematics and computer science. Originating from chess, the puzzle has intrigued mathematicians since its introduction in the mid-19th century. The challenge of placing N queens on an N×N chessboard without any of them attacking each other encapsulates the essence of combinatorial problem-solving.

1.2 Mathematical Underpinnings

The N-Queens puzzle can be framed mathematically as an optimization problem. Given an N×N chessboard, the objective is to find a placement of N queens such that no two queens share the same row, column, or diagonal. This seemingly simple problem becomes increasingly complex as N grows, with a solution requiring careful consideration of the chessboard's structure and potential configurations.

1.3 Inherent Difficulty

The inherent difficulty of the N-Queens problem arises due to its combinatorial nature. The total number of possible configurations grows rapidly with the chessboard size, leading to an exponential increase in the solution space. As a result, finding a solution becomes a non-trivial task, necessitating the application of efficient algorithms for exploration and optimization.

CHAPTER 2

TOOLS & METHODOLOGY

2.1 Programming Language - Python

Python is chosen as the programming language for the N-Queen Visualizer due to its readability, versatility, and extensive support for libraries and frameworks. The clean syntax of Python facilitates the implementation of complex algorithms while ensuring the code remains accessible to a broad audience, including those new to programming.

2.2 Graphics Library - Pygame

Pygame is selected as the graphics library for its suitability in creating interactive applications. Pygame simplifies the process of handling graphical elements, user input, and real-time updates. Its ease of use makes it an ideal choice for developing a visually engaging tool like the N-Queen Visualizer.

2.3 Backtracking Algorithm

The backtracking algorithm is at the core of solving the N-Queens problem. This algorithm systematically explores possible configurations, making informed decisions at each step to prune the solution space. The decision-making process involves placing queens on the chessboard and backtracking when conflicts arise. The visualization of the algorithm provides users with a dynamic representation of the solution-finding process.

CHAPTER 3

RESULTS & DISCUSSION

3.1 Visual Representation

The N-Queen Visualizer effectively translates the abstract concept of the N-Queens puzzle into a dynamic and visually comprehensible representation. For each N value input by the user, the visualizer generates real-time solutions, displaying the placement of queens on the chessboard. The graphical interface, powered by Pygame, vividly illustrates the steps taken by the backtracking algorithm, showcasing the process of exploration, decision-making, and solution space pruning.

3.2 Effectiveness of the Backtracking Algorithm

The results unequivocally demonstrate the efficacy of the backtracking algorithm in solving the N-Queens problem. Users can observe how the algorithm intelligently navigates through the chessboard, placing queens strategically and backtracking when conflicts arise. The visual cues, such as green nodes indicating the current exploration and red lines marking paths with existing queens, contribute to a clear understanding of the algorithm's decision-making process.

3.3 Impact of N on Solution Complexity

The discussion delves into the impact of varying N values on the complexity of finding solutions. As N increases, the number of possible configurations grows exponentially, posing a greater challenge for the algorithm. The visualizer allows users to observe how the algorithm adapts to different N values, providing valuable insights into the scalability and efficiency of the backtracking approach.

3.4 Enhanced User Interaction

Embark on the N-Queens Journey with the "Start" Feature

Engage in the thrilling exploration of the N-Queens puzzle by initiating the search with the "Start" functionality. Witness the backtracking algorithm in action as it intelligently navigates the chessboard, unveiling solutions in real-time.

Seamless Control: Retry for a Fresh Attempt, Quit for a Graceful Exit

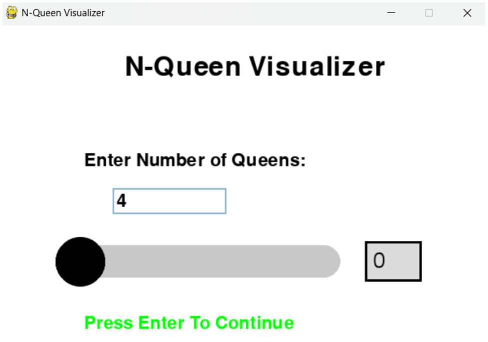
Empower your interactive experience with the added functionalities. Retry seamlessly takes you back to the input box, encouraging experimentation. Meanwhile, Quit gracefully concludes the game, providing users with a convenient exit when their exploration is complete.

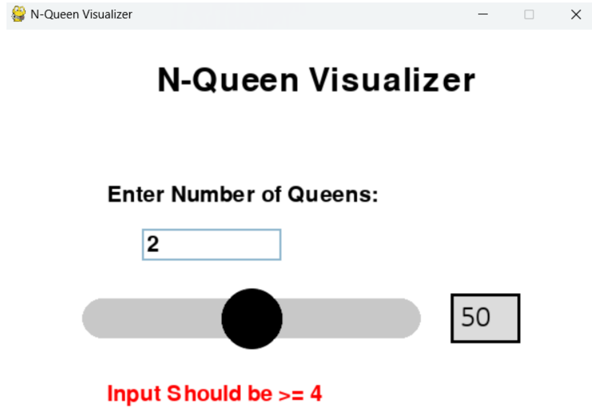
3.5 Symbolism and Representation

Delve into the color-coded visual elements employed in the N-Queen Visualizer, providing an in-depth explanation of their meanings. Explore how red signifies spots where queens are already placed, green indicates the algorithm's current spot-checking process, and blue represents the condition-checking boxes, offering users a clear and intuitive understanding of the backtracking algorithm's dynamics.

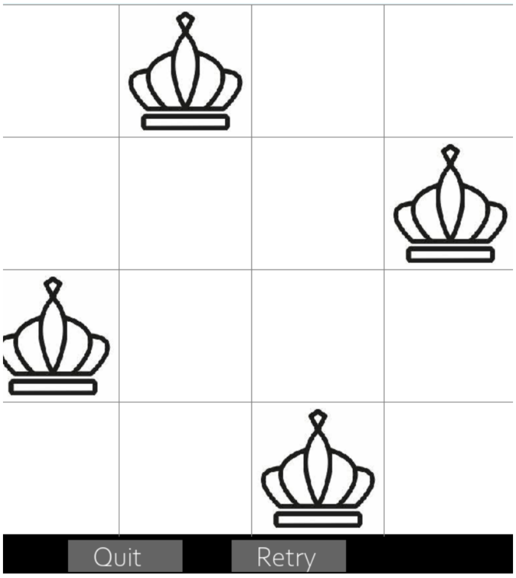
3.6 User Feedback and Observations

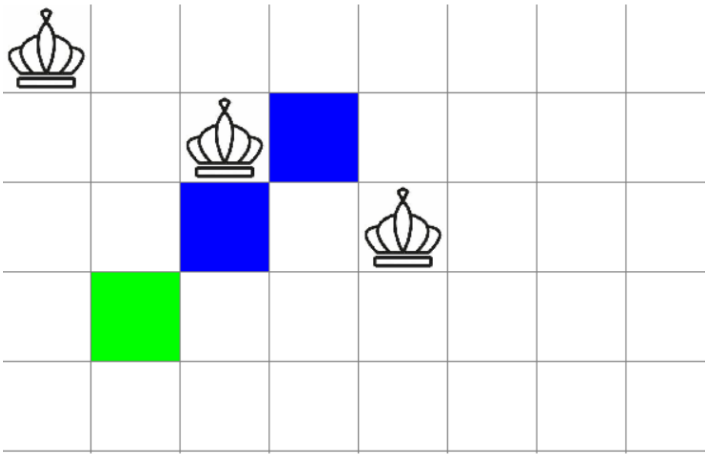
User feedback plays a crucial role in the evaluation of the visualizer. Insights gathered from user testing sessions, comments, and observations contribute to the refinement of the tool. Common user experiences, challenges encountered, and suggestions for improvement are considered in the ongoing development of the visualizer. This iterative process ensures that the tool remains user-friendly and effective in conveying the nuances of the N-Queens problem.


N-Queen Visualizer
N-Queen Visualizer
Enter Number of Queens:
4
0
Press Enter To Continue


N-Queen Visualizer
N-Queen Visualizer
Enter Number of Queens:
2
50
Input Should be >= 4


Quit
Retry
Start


Quit
Retry



CHAPTER 4

FUTURE WORK

4.1 Additional Features

This section explores potential avenues for enhancing the N-Queen Visualizer. Ideas for additional features, such as different color schemes for visual elements, customizable speed settings, or alternative algorithms for solving the N-Queens problem, are discussed. Incorporating these features could further enrich the user experience and provide a more comprehensive educational tool.

4.2 User Interaction Improvements

Future work may focus on refining the user interface to enhance user interaction. This could involve incorporating tooltips, informative pop-ups, or interactive elements that allow users to pause, resume, or interact with the algorithm in real-time. Improving the overall usability and accessibility of the visualizer ensures a more engaging and educational experience.

4.3 Exploration of Alternative Algorithms

While the backtracking algorithm forms the core of the N-Queen Visualizer, exploring alternative algorithms could broaden the tool's educational scope. Consideration of different algorithms, such as genetic algorithms or constraint satisfaction approaches, could offer users a more comprehensive understanding of diverse problem-solving strategies.

4.4 Performance Optimization

Optimizing the performance of the visualizer is a continual consideration. This involves refining the codebase, implementing efficient data structures, and exploring ways to handle larger N values without compromising the user experience. Performance enhancements ensure that the visualizer remains a responsive and scalable tool.

4.5 Educational Modules

Expanding the educational value of the visualizer may involve incorporating tutorial modules, explanatory text, or interactive quizzes to reinforce key concepts. Providing users with additional resources and information enhances the overall learning experience and encourages exploration.

In conclusion, Chapter 4 outlines potential directions for future development, ensuring that the N-Queen Visualizer remains a dynamic and evolving tool with continuous educational value. Consideration of additional features, user interaction improvements, exploration of alternative algorithms, performance optimization, and the incorporation of educational modules contribute to the ongoing refinement of the visualizer.