ASE Group (6)

Web Development journey of N-Queens and Polysphere Puzzle Project

Abstract

Our journey in developing the N-Queens and Polysphere Puzzle web application has been a challenging yet rewarding experience. As a team of eight members, we were tasked with creating a dynamic and engaging platform that showcased both the N-Queens puzzle and the Polysphere Puzzle. This project required a diverse set of skills, ranging from backend development and frontend design to quality assurance testing and documentation. In this reflective essay, I will delve into the collaborative efforts, individual contributions, challenges faced, and the lessons learned throughout the development process.

Duties and responsibilities of each group member:

Use the data structure to store it in DB	: **Jawad, Tom, Aman, Rohan**
Front-end Design like web-based game rather than a web page	: **Junkai, Tom, Aman, Rohan**
New Project Plan for this Task	: **Tom, Aman, Rohan**
Timeline Chart for this project	: **Jawad**
Search algorithm and combining Front-end and Back-end	: **Aman, Tom, Rohan**
Research what is Github Workflows, CI/CD Tools	: **All**
QA	: **Rimjim, Veerpal, Mya**
Documentation:	: **Rimjim, Veerpal, Mya**

Introduction

Both "N-Queens" and "Kanoodle" solver are puzzle games, however they belong to different puzzle categories. N-queen is one of the combinatorial puzzles, also known as chess puzzles in this context, entail arranging or picking items based on certain rules or limitations. Chess puzzles are concerned with the positioning and movement of the pieces of chess on a chessboard. The goal of the N-Queens puzzle is to arrange N chess queens on a NxN chessboard so that no two queens threaten each other. This means that no two queens can be in the same row, column, or diagonal at the same time. The task is to identify all potential configurations that satisfy these constraints.

Among any other Pattern Matching Puzzles, Kanoodle is placing a set of 11 pieces into a three-dimensional puzzle board. The pieces have different forms, and the goal is to arrange them so that they fit together. Pattern Matching Puzzles frequently need spatial reasoning, vision, and strategic thinking. Identifying and managing patterns in a spatial context is required to solve them. Arranging parts to match a precise pattern, building structures, or accomplishing objectives within a limited space can all be challenging.

Collaborative Efforts

The eight-person group project for Polysphere Puzzle and N-Queens web development was a dynamic process that demonstrated the power of cooperation in overcoming obstacles. Since half of the group had only a medium level of programming experience, our approach was to create a welcoming atmosphere that promoted ongoing education and teamwork. Frequent knowledge-sharing meetings were started, where team members with greater programming experience guided and supported team members with less experience. This strategy strengthened the team's less seasoned members' abilities while also fostering a sense of camaraderie.

A distributed approach to tasks recognized the group's diverse skill set and proved beneficial for the project. Team members with medium-level knowledge contributed to the development process by taking on specific tasks and functionalities, while those with stronger programming backgrounds tackled more complex coding challenges. Because of this division of labor, each person was able to use their skills and expertise to meaningfully contribute to the project. Collaborative coding sessions were also arranged, offering a chance for knowledge sharing and real-time collaboration. For team members with intermediate programming experience, this method not only sped up the learning curve but also produced a more unified and well-integrated codebase.

Making sure that everyone was on the same page regarding the project's objectives required effective communication. To promote open communication, regular team meetings, status reports, and open forums for problem-solving were set up. Feedback loops were promoted in Github because they let team members exchange problems and insights and promoted a continuous improvement culture. Although working together wasn't without its difficulties, the group's dedication to helping one another and utilizing individual talents led to a supportive and effective atmosphere. Ultimately, the project's accomplishment was a tribute to the strength of teamwork and the combined endeavor of a group of programmers with varying levels of expertise striving for a shared objective.

Backend Development

Three team members took on the challenge of backend development, tackling the logic behind the N-Queens puzzle and Polysphere Puzzle algorithms. The backend developers faced the task of ensuring the puzzles' functionality, scalability, and responsiveness. The collaborative efforts of the backend team resulted in a robust and efficient system that seamlessly handled the puzzle-solving logic.

A result of restricted computer resources, the N-Queens conundrum, renowned for its computational complexity, needed a resource-efficient solution. It turned out that the Dancing Link algorithm X was the best option. In our situation, this algorithm produced the 724 correct solutions for the 10-queens chess board with efficiency. It is particularly good at solving precise cover issues. The time needed to find the solution set was greatly decreased by its capacity to retrace its steps and make educated decisions about which paths are the most promising.

Fine-tuning parameters and making sure the algorithm worked with the limited resources we had available were two of the difficulties we faced when optimizing the method. Working together, our backend team experimented with various configurations to find a solution that balanced accuracy with speed of execution. The N-Queens puzzle's successful use of the Dancing Link algorithm X demonstrated how crucial it is to select the appropriate algorithm for the given problem area.

A 2D version of the Kanoodle puzzle called the Polysphere Puzzle had its own set of difficulties. The puzzle required a strategy to reduce the number of possible answers to 80,445 while taking resource limitations into account. It was vital that we forwent the extra rotations and flips that would have greatly increased the time complexity.

By putting the Polysphere Puzzle's Dancing Link algorithm X into practice, we were able to quickly navigate the solution space and take into account only the most crucial configurations. The algorithm was a perfect fit for our puzzle because of its flexibility and capacity to deal with limits. The backend team thoroughly examined the limitations of the puzzle to make sure the selected method complied with the challenge criteria and minimized the collection of possible solutions.

Substantial challenges were faced during the backend construction of the Polysphere and N-Queens puzzles, but creative solutions were found. To optimize our computations and produce results within the allotted time frames, the Dancing Link algorithm X was strategically implemented. This project demonstrated our backend development team's technical proficiency while also emphasizing the significance of choosing the appropriate algorithm for certain problem domains and resource limitations. As we think back on our adventure, we see how important it is to be flexible, work with others, and make calculated decisions in order to overcome difficult computational problems.

Frontend Development

Two team members dedicated their efforts to crafting an intuitive and visually appealing frontend for our web application. Their responsibilities included designing the user interface, implementing responsive design principles, and ensuring a seamless user experience. The frontend developers worked closely with the backend team to integrate the puzzle-solving algorithms into the user interface, creating a cohesive and engaging platform.

Quality Assurance Testing and Documentation

Our team assigned two group members to handle the vital responsibilities of documentation and quality assurance testing. The job of the QA testers was to find and fix any defects or problems in the application so that the user experience was seamless and error-free. Concurrently, the documentation team produced an invaluable resource for future reference by carefully recording every step of the development process, from original design to final implementation.

1. N-Queens Web Application

In order to ensure the correct operation and precision of the N-Queens web application, which is especially made for a 10-queens chess board with 724 solutions while taking into account the limitations of computational resources.

1.1 Solution Count

Check that 724 solutions are generated correctly for a 10-queens chess board by the N-Queens application. Verify that the solutions follow the chess rules, which prohibit having two queens in the same row, column, or diagonal.

1.2 Performance

Run the application within the specified computational resource limits to evaluate its performance. confirm that the program stays stable and responsive throughout the computation process.

1.3 Error Handling

Access some intentional error in the input parameters (such as going over the size of the chess board), and make sure the program handles them nicely. Check to see if the program clearly displays error messages when input is incorrect.

1.4 Compatibility

To make sure the application is cross-browser compatible, test it using various web browsers. Validate that the user interface works well and responds to different screen sizes.

2. Polysphere Puzzle Web Application

To verify the functionality and efficiency of the Polysphere Puzzle web application, focusing on a reduced solution set of 80445 solutions using the Dancing Link Algorithm X.

2.1 Solution Set

Confirm that the Polysphere Puzzle application generates 80445 solutions after implementing the Dancing Link Algorithm X. Ensure that the reduced solution set maintains the integrity of the original puzzle.

2.2 Performance

Test the application's performance by measuring the time taken to generate the reduced solution set. Ensure that the time complexity is within acceptable limits (39 minutes or less).

2.3 Rotation and Flip Handling

Verify that the removal of additional rotations and flips does not affect the accuracy of the puzzle solutions. Confirm that the puzzle pieces are correctly positioned according to the new constraints.

2.4 Error Handling

Support some inappropriate errors in the input parameters and validate that the application provides appropriate error messages. Ensure that the application gracefully handles unexpected errors during the solution generation process.

Check to see if both applications' user interfaces are simple to use. Review that the generated solutions are clearly displayed and that users can interact with them with ease. Consult the project documentation to ensure that it contains comprehensive instructions on how to use the applications. Ensure that the documentation reflects any changes made throughout the development process. Ensure that the front-end and back-end components function well together to offer a smooth user experience. Verify that changes made to one part don't affect how the other works.

Challenges and Lessons Learned(Need to add more experiences by each member)

Participating in the 8-member group project for N-Queens and Polysphere Puzzle web development presented a unique set of challenges, particularly due to the diverse skill levels within the team. Half of the members lacked sufficient programming language expertise and familiarity with tools such as Django, Agile methodologies, and GitHub. This posed a hurdle in achieving a seamless workflow, as collaboration heavily relies on a shared understanding of these technologies. Overcoming this challenge required an emphasis on mentorship and knowledge sharing. The team initiated regular learning sessions, allowing members to grasp the basics of programming languages, the Django framework, and version control using GitHub. Incorporating Agile practices was also a learning curve, as it necessitated adapting to iterative development cycles and continuous feedback. The challenges posed by varying skill levels ultimately fostered a culture of collaboration and support, as more experienced members took on mentorship roles, resulting in a more inclusive and knowledgeable team. In final results, the difficulties brought about by different skill levels promoted a culture of cooperation and support as more seasoned team members assumed mentorship responsibilities, creating a more diverse and knowledgeable group.

Participating in the web development project for N-Queens and Polysphere Puzzle provided invaluable insights into the value of good communication and the necessity of flexible project management techniques. Since team members' levels of expertise varied, it became crucial to communicate in an open and transparent manner. It was decided to have regular check-ins and feedback sessions to make sure that everyone was in agreement and that any doubts or challenges were resolved right away. Furthermore, the implementation of Agile methodologies demonstrated significant value in effectively handling the dynamic project specifications and accommodating team members' learning curves. Classifying flexibility in project management enabled modifications according to personal learning rate and guaranteed that project objectives persisted. With the benefit of the future, this project has highlighted the importance of creating an inclusive workplace where knowledge sharing is valued and integrated into teamwork, resulting in a more capable and resilient group.

Conclusion

The web development project of N-Queens and Polysphere Puzzle, which involved an 8-member team with varying programming skills, is evidence of the effectiveness of teamwork, flexibility, and shared learning. The team's diverse skill set presented some initial challenges, but our combined efforts and dedication to sharing knowledge have paid off, as the project has turned out well. The experience has shown how crucial it is to create a welcoming atmosphere where team members with varying degrees of programming expertise are motivated to actively participate and pick up tips from colleagues with greater experience.

We have seen personally the transformational power of adopting Agile methodologies and maintaining consistent communication on Github throughout this journey. Frequent check-ins and feedback loops were crucial for promptly resolving issues and coordinating our objectives. We were able to adapt to the medium programming knowledge of team members due to the flexibility innate in Agile practices, making sure that everyone felt empowered to make a significant contribution to the project. Ultimately, our combined efforts produced a working N-Queens and Polysphere Puzzle web application and improved the team's collective expertise. This project serves as an example of the potential of inclusive teamwork, where people with different backgrounds and skill sets collaborate to overcome obstacles and realize a common goal in web development.