

|  |  |
| --- | --- |
| **Title** | Reflective Essay by Group 6 |
| **Candidate Numbers** | 276187, 276267, 276280, 276293, 276375, 277822, 284318, 284946 |
| **Module Leader** | Prof. Hsi-Ming Ho |
| **Course Title** | MSc Advanced Computer Science |
| **Date Submitted** | 12/15/2023 |

947G5: Advanced Software Engineering

**Abstract**

Our journey in developing the Puzzles web application encounters the challenging and rewarding experience. As a team of eight members, we were assigned to create different kinds of Puzzles in web platform. 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 describe the process of each tasks and challenges faced by each team member throughout the development process.

**Table of Contents**

[1 Introduction 1](#_Toc153158871)

[2 Collaborative Efforts 2](#_Toc153158872)

[3 Backend Development 3](#_Toc153158873)

[3.1 Task 2 (N-Queens puzzle) 3](#_Toc153158874)

[3.2 Task 3 (Polysphere puzzle) 3](#_Toc153158875)

[3.3 Task 4 (Polysphere pyramid puzzle) 4](#_Toc153158876)

[3.4 Task 5 (Freedom Task) 4](#_Toc153158877)

[4 Frontend Development 5](#_Toc153158878)

[5 Quality Assurance Testing and Documentation 5](#_Toc153158879)

[5.1 N-Queens (Task 2) 5](#_Toc153158880)

[5.2 Polysphere Puzzle (Task 3) 6](#_Toc153158881)

[5.3 Polysphere Pyramid Puzzle (Task 4) 6](#_Toc153158882)

[6 Challenges and Lessons Learned 7](#_Toc153158883)

[7 Conclusion 8](#_Toc153158884)

# Introduction

Developing the puzzles through collaborative efforts on GitHub is the main task of the group project, all team members were contributed to developing N-Queens puzzle, Polysphere Puzzle, Polysphere Pyramid Puzzle, and additional experiment.

In the task 1, all the members must create GitHub accounts so that each member can contribute to each task.

As a task 2, the N-Queens puzzle involves arranging N chess queens on an NxN chessboard by placing each chess queens in the terms of avoiding threats between each queen in the same row, column, or diagonal. The task focus on identifying all possible solutions by using the appropriate and effective algorithms.

Moving on to Task-3, the primary objective was to develop Polysphere puzzle by using the algorithms and different programming languages. Among other Pattern Matching Puzzles, the nature of Polysphere puzzle is placing 11 different shaped pieces into a three-dimensional puzzle board. Players must think about the angle and the shape of the pieces before solving the puzzle. The challenge involves arranging pieces to fit together, choosing the right angle, and placing the pieces to get the completed puzzle board.

Another development of "Polysphere Puzzle" into a pyramid structure was a part of the group project. This task aimed to design a professional puzzle game where players manipulate and solve puzzles with pyramid-shaped structures and three-dimensional spherical objects. Challenges may involve adjusting or placing the different pieces within a three-dimensional space to get a pyramid structure.

In Task-5, the team decided to create a user-friendly webpage with all the developed puzzles. The main goal of this project is to find out the possible solutions from individual puzzles, presenting the team's collective knowledge and collaborative abilities. The website tends to pursue the players and web developers to improve their respective skills.

# Collaborative Efforts

|  |  |
| --- | --- |
| **Duties & Responsibilities** | **Members** |
| 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** |

The eight-members project group displayed the efficiency of collaborative teamwork in overcoming challenges. In task-1, each team members must create GitHub account to interact with other team members and contribute in the project. It was quite challenging for some of the team members to access the GitHub.

To support team members with various level of programming experiences, regular knowledge-sharing meetings were arranged by zoom to achieve the mentorship from more experienced members and encouraging each member to contribute in the different roles to get the experience of the actual workplace. This group project not only sped up the learning curve but also produced a well-integrated codebase for team members with intermediate programming knowledge.

Ensuring alignment on project deliverables required effective communication, regular team meetings, updating the changes of each progress in discussion panel, sharing and solving different issues, and collecting different ideas from each team members on GitHub. Tracking feedback loops on GitHub allowed for the exchange of insights and continuous improvement. Despite challenges, the group's commitment to mutual support and encouraging individual talents to build a cohesive and effective atmosphere during the progress of each tasks. Every single change by each team members are always updated on GitHub so that everyone can know the issues and the requirements to finish each task. As the project's success depends on the strength of teamwork among programmers with diverse experience, team leader and experienced members are always checking and supporting each member to contribute in the project and to finish their respective roles in time.

# Backend Development

## Task 2 (N-Queens puzzle)

* Three experienced team members took a responsibility on backend development for N-Queens Puzzle.
* Tasks included looking for the algorithms and logic and backend process.
* Collaborative teamwork resulted in a robust and efficient system for puzzle-solving logic.
* Due to restricted computer resources, an efficient solution was needed for the computationally complex N-Queens puzzle.
* The Dancing Link Algorithm X was chosen for its efficiency, producing 724 correct solutions for the chessboard size from 4x4 to 10x10.
* Getting a permission to access limited resources was one of the challenges faced by the backend team.
* Frequent QA testing after each change makes the solutions in the puzzle to precise.
* Successful use of the Dancing Link algorithm X for N-Queens highlighted the importance of selecting the right algorithm for a given problem.

## Task 3 (Polysphere puzzle)

* Implemented Donald Knuth’s Algorithm X for exact cover problem to solve our Polysphere puzzle rather than primitive brute force techniques. The algorithm efficiently produced 80,444 solutions within a duration of 39 minutes.
* The solutions are based on the actual and mirror sets of combinations, filtering out unnecessary combinations that brute force algorithm generated naturally which could have affect the time complexity of the puzzle.
* A use-friendly web UI has been integrated for seamless gaming experience using CSS, HTML and Jscript to create the puzzle.
* App has been also hosted in the cloud for people who don’t want to host the game inside their local machines.
* Django Framework has been used and the architecture followed in MVT.
* Project Files can be accessed inside the Polysphere directory for the Polysphere Puzzle.
* Various bugs were encountered during the development of this version which can be seen in the Issues tab of ASE Group 6 GitHub Repository.

## Task 4 (Polysphere pyramid puzzle)

Some additional experiments will be drawn such as

* Implement Donald Knuth Algorithm X solutions yet to be verified.
* Will be using three.js and orbitcontrol.js to render our 3D space for polysphere pyramid puzzle.
* Interactive UI for user to play the game.
* Django framework used and MVT architecture followed.

## Task 5 (Freedom Task)

The following tasks will be added.

* Fixing any errors posted by our QA team in GitHub issue.
* Combining all task into one complete web application that will be hosted in the web for people of all ages to play.
* Probably integrate username and password for people to have their own account to play and save their progress.
* Introduce levels for all the puzzles.
* Keep the repo updated of any feature issue or new request handle by our team.

# Frontend Development

Two team members focused on crafting an intuitive and visually appealing frontend for the web application. Their responsibilities are

* Designing the user interface and implementing responsive design principles.
* Ensured a polished user experience through collaborative efforts with the backend team.
* Worked closely with backend developers to integrate puzzle-solving algorithms into the user interface.
* Aimed to create a cohesive and engaging platform for users.

# Quality Assurance Testing and Documentation

Two team members were assigned documentation and quality assurance testing responsibilities, with the QA testers tasked to identify and rectify application defects for a seamless user experience. Issues were reported on GitHub, and team members identified and implemented solutions.   
Documenting issues and experiences on GitHub to prepare for the final document.

## N-Queens (Task 2)

* Verify the N-Queens web application generates 724 solutions correctly for a 10-queens chess board.
* Run the application within specified computational resource limits.
* Evaluate the performance to ensure stability and responsiveness throughout the computation process.
* Intentionally introduce errors in input parameters, such as exceeding the chess board size and placing the queens in the wrong places on the chess board.
* Confirm the program handles errors gracefully, displaying clear and informative error messages.
* Validate that the user interface functions correctly and responsively on different screen sizes and resolutions.

## Polysphere Puzzle (Task 3)

* Verify functionality and efficiency of Polysphere Puzzle web application with a focus on a reduced solution set of 80,445 (i.e., discarding rotations and reflections) solutions using Dancing Link Algorithm X.
* Confirm the generation of 80,445 solutions by the Polysphere Puzzle application post-implementation of Dancing Link Algorithm X.
* Ensure that the reduced solution set maintains the integrity of the original puzzle configurations.
* Test application performance by measuring the time taken to generate the reduced solution set, ensuring time complexity is within acceptable limits (39 minutes or less).
* Validate that removal of additional rotations and flips does not impact the accuracy of puzzle solutions, and pieces are correctly positioned based on new constraints.
* Support inappropriate errors in input parameters and verify that the application provides appropriate error messages.
* Ensure the application gracefully handles unexpected errors during the solution generation process.

## Polysphere Pyramid Puzzle (Task 4)

* Implement and verify the Dancing Link Algorithm X for the Polysphere Pyramid Puzzle.
* Reduce time to load and resource consumption while optimizing web development to create an agile and responsive gaming experience that improves user satisfaction overall.
* Provide accurate and intuitive controls to rotate and flip the transparent sphere so that players can interact with the puzzles naturally and explore various viewpoints with ease.
* Identify potential input errors and provide clear, informative error messages to guide users.
* Address any compatibility issues to provide a consistent user experience across platforms.
* To handle unexpected user inputs or system glitches effortlessly and maintain the game's stability and usability throughout the experience, implement efficient error handling mechanisms.

# Challenges and Lessons Learned

Participating in the 8-member group project for web development of puzzles presented a unique set of challenges. Particularly, due to the diverse skill levels within the team.

Half of the team possessed limited programming language expertise and were unfamiliar with tools such as Django, Agile methodologies, and GitHub. In the beginning of the project, it was quite difficult to get a seamless workflow, as collaboration heavily relies on a shared understanding of the technology. The team initiated regular learning sessions, allowing members to improve the knowledge of programming languages, the Django framework, and version control using GitHub. The incorporation of Agile practices presented a learning curve, accessing the establishment of regular check-ins and feedback sessions to ensure consensus and the timely resolution of doubts. 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.

Substantial challenges were faced during the backend construction of all the tasks, but creative solutions were found by using the effective algorithms. To optimize our computations and produce results within the limited time frames, the Dancing Link algorithm X was strategically implemented. As we think back on our adventure, we see how important it is to be flexible to work with others and make decisions to overcome difficult computational problems.

# Inspirations

Task 1:

Task 2:

Task 3: <https://kanoodlesolver.vercel.app/>

<https://gist.github.com/anonymous/75281bc0b5063453126c>

<https://github.com/search?q=kanoodle&type=repositories>

Task 4: <https://github.com/benelder/kanoodle-3d>

<https://github.com/wkeeling/kanoodlegenius2d>

Task 5: <https://threejs.org/examples/#webgl_camera_array>

<https://threejs.org/examples/webgl_multiple_elements.html>

# Conclusion

Our eight-person ASE Group has found the web development process for puzzles is both rewarding and challenging. Each task required a different skill set and teamwork from solving the complex N-Queens puzzle to creating the additional web development. The team's dedication to inclusive practices such as frequent mentorship and learning sessions proved crucial in overcoming the problems presented by disparate skill levels. Effective communication and agreement were ensured by implementing Agile practices, check-ins, and feedback sessions. Using sophisticated algorithms such as the Dancing Link Algorithm X highlighted the significance of strategic decision-making in solving computational difficulties. As we reflect on our experience, it's clear that cooperative efforts, flexibility, and well-considered decision-making are essential for web development projects to succeed. This experience has improved our technical proficiency and highlighted the importance of a welcoming and cooperative work environment.