**ASE Group (6)**

**Web Development journey of puzzle games**

**Abstract**

Our journey in developing the Puzzles 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 the Puzzles. 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**

As developing the puzzles through collaborative efforts on GitHub is the main task of the group project, all team members were contributed in developing N-Queens puzzle, Kanoodle Puzzle, Polysphere Pyramid Puzzle, and additional experiment. As a combinatorial puzzle rooted in chess, the N-Queens puzzle involves arranging N chess queens on an NxN chessboard, adhering to the constraints of avoiding threats between queens in the same row, column, or diagonal. The task centered on identifying all potential configurations meeting these constraints.

Moving on to Task-3, the primary objective was to develop Kanoodle puzzles, leveraging algorithms and diverse programming languages. Kanoodle, categorized among Pattern Matching Puzzles, entails placing 11 differently shaped pieces into a three-dimensional puzzle board, requiring spatial reasoning, vision, and strategic thinking. The challenge involves arranging pieces to fit together, emphasizing pattern recognition and achieving objectives within a confined space.

The subsequent task involved creating the "Polysphere Pyramid Puzzle" by developing the Kanoodle puzzle into a pyramid structure. This task aimed to design a game where players manipulate and solve puzzles with pyramid-shaped structures and three-dimensional spherical objects. Challenges may involve aligning or assembling pyramid pieces within a three-dimensional space, demanding spatial reasoning, pattern recognition, and logical thinking.

In Task-5, the team unanimously decided to amalgamate all the developed puzzles into a comprehensive puzzle game website. The overarching objective of this project is to integrate solutions from individual puzzles, showcasing the team's collective knowledge and collaborative abilities. The website aspires to be an elegant and captivating platform, emphasizing the team's proficiency in puzzle development and teamwork.

**Collaborative Efforts**

The eight-person web development puzzle project showcased the efficacy of collaboration in overcoming challenges. To support team members with intermediate programming experience, a welcoming atmosphere was cultivated, emphasizing continuous education and teamwork. Regular knowledge-sharing meetings facilitated mentorship from more experienced members, enhancing skills and fostering a sense of fellowship among the team.

Adopting a distributed approach to tasks, the group leveraged its diverse skill set effectively. Team members with intermediate knowledge undertook specific development tasks, while those with advanced programming skills addressed complex coding challenges. This division of labor allowed each member to contribute meaningfully based on their skills. Collaborative coding sessions facilitated real-time collaboration and knowledge sharing, expediting the learning curve for team members with intermediate programming experience and resulting in a cohesive and well-integrated codebase.

Ensuring alignment on project objectives necessitated effective communication, facilitated through regular team meetings, status reports, and open problem-solving forums on Github. Leveraging feedback loops on Github allowed for the exchange of insights and continuous improvement. Despite challenges, the group's commitment to mutual support and leveraging individual talents fostered a cohesive and effective atmosphere. The project's success stands as a testament to the strength of teamwork among programmers with diverse expertise striving for a shared objective..

**Backend Development**

**Task 2 (Nqueens puzzle)**

* Three team members took on backend development for N-Queens and Polysphere Puzzle algorithms.
* Tasks included tackling logic, ensuring functionality, scalability, and responsiveness.
* Collaborative efforts 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 10-queens chess board.
* The algorithm is adept at solving precise cover issues and efficiently retracing steps for educated decisions.
* Fine-tuning parameters and optimizing for limited resources were challenges faced by the backend team.
* Experimentation with various configurations was done to balance accuracy with speed of execution.
* 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 (Kanoodle puzzle)**

* Implemented Donald Knuth’s Algorithm X for exact cover problem to solve our Kanoodle puzzle rather than primitive brute fore techniques.The algorithm efficiently produced 80,444 solutions within a duration of 39 minutes.
* The 80,444 solutions are based on the actual and mirror sets of combinations, flitering out unnecessary combinations that brute force algorithm generated naturally which could have affect the time complexity of the puzzle.
* A user friendly web UI has been integrated for seamless gaming experience using CSS, HTML and Jscript to dynamify 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 Kanoodle 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 experiements 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.
* introuduce 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 seamless 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.

**Kanoodle Puzzle (Task 3)**

* Verify functionality and efficiency of Kanoodle Puzzle web application with a focus on a reduced solution set of 80,445 solutions using Dancing Link Algorithm X.
* Confirm the generation of 80,445 solutions by the Kanoodle 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.
* In order 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. This posed a hurdle in achieving a seamless workflow, as collaboration heavily relies on a shared understanding of these technologies. 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, necessitating 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. 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, work with others, and make calculated decisions in order to overcome difficult computational problems. This project underscores the significance of fostering an inclusive workplace that values knowledge sharing, integrating it into teamwork for enhanced capability and resilience.

**Conclusion**

Our eight-person ASE Group has found the web development process for puzzles to be both rewarding and challenging. Each task required a different skill set and teamwork, from solving the complex N-Queens puzzle to creating the Kanoodle and Polysphere Pyramid puzzles and additional web development. The team's dedication to inclusive practices—such as frequent mentorship and learning sessions—proved crucial in overcoming the obstacles 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 surmounting 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, both of which have contributed to the project's overall success.