School of Engineering and Informatics

UNIVERSITY OF SUSSEX

947G5: Advanced Software Engineering

Title	Navigating the Landscape: Reflections on our Software Development Journey
Project Group	6
Assignment Type	Reflective Essay
Candidate Names & Numbers	Tom Naccarato - 276187 Jawad Nasar Shah - 276267 Rohan Kadam - 276280 Veerpal x - 276293 Rimjhim Chakraborty - 276375 Aman Thapa Magar - 277822 Mya Lwin - 284318 Junkai Ou - 284946
Module Leader	Prof. Hsi-Ming Ho
Course Title	MSc Advanced Computer Science
Date Submitted	15 December 2023

Abstract

Our journey in developing the puzzles web application was a challenging and rewarding experience. As a team of eight members, we were assigned to create different kinds of puzzles in web platform. This coursework has helped us gain insights on industrial approach as to how products are designed, developed, and delivered. 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, we will describe the process of each task and challenges faced by each team member throughout the development process.

Table of Contents

Table of Figures

1 Introduction

Software development teams has been a norm in many organizations around the world. A group of experts working towards the same vision can bring about solutions faster than individual efforts. It is essential to point out the fact that not every team can have success as factors such as personality, vision, passion, and motivation towards a common goal is very rare to achieve as you need the right balance to be a successful team. Hence, the same thing can be said about our team developing puzzles through collaborative efforts which was the main objective of this project. By using a collaborative platform such as GitHub, all the team members were able to make contributions in developing the puzzle games. The development of these games was a unique experience as each one of them had various learning objectives which was delivered using practical approach. Each group members were assigned various tasks which led to delivering insightful applications such as N-Queens Puzzle, Polysphere Puzzle and Polysphere Pyramid Puzzle. At first glance, we summarised the project to be a fun-involved development but slowly understood that these games were the backbone of how computations, logics, space-time complexity and efficiency of algorithms can be measured which was an insightful experience as we understood at the very depths of what a software engineer should aspire to be; introduce a bit of entertainment in your productivity. We drew conclusions that a team should have the right balance of entertainment and productivity which were directly proportional to the tasks we deliver.

The group of eight members were assigned various tasks at hand which were subsidiaries of the main deliverables:

- **Task 1:** Creating GitHub account for collaboration and contributions
- **Task 2**: Development of N-Queens Puzzle
- **Task 3:** Development of Polysphere Puzzle
- **Task 4:** Development of Polysphere Pyramid Puzzle
- **Task 5**: Freeform Development

The team was divided into Team Leader, Overseer, Frontend, Backend, QA, and Development. During the project timeline, there were many setbacks and hurdles faced which was a learning curve for the team overall. Therefore, this essay will discuss about key points on teamwork, technical challenges and lessons learned.

2 Team Dynamics

2.1 Team Overview

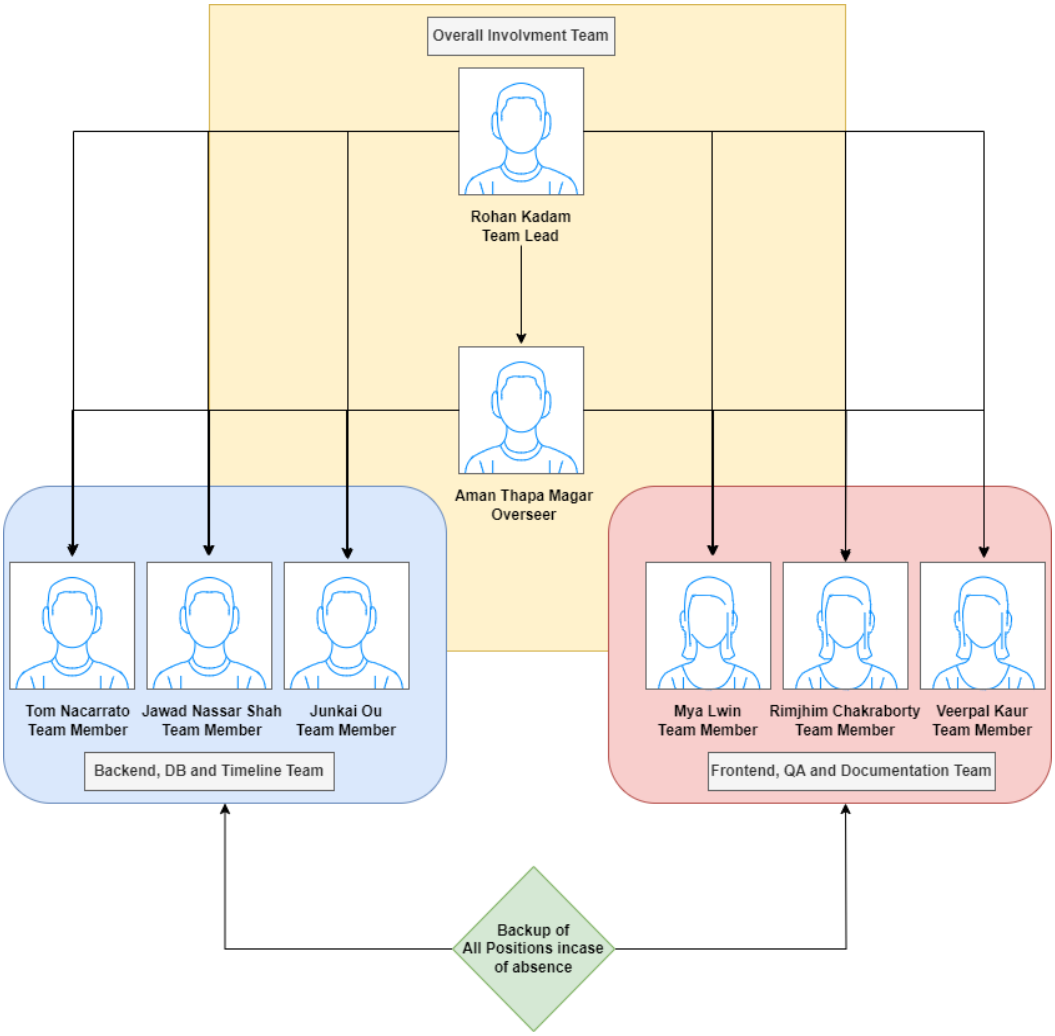

Figure 1: Team Division Chart

The team displayed the efficiency of collaborative teamwork in overcoming challenges. 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 encouragement built a cohesive and effective atmosphere during the progress of each task. As the project's success depends on the strength of teamwork, experienced members were always checking and supporting other members to contribute to the project and to finish their respective roles in time.

2.2 Timeline and Operations

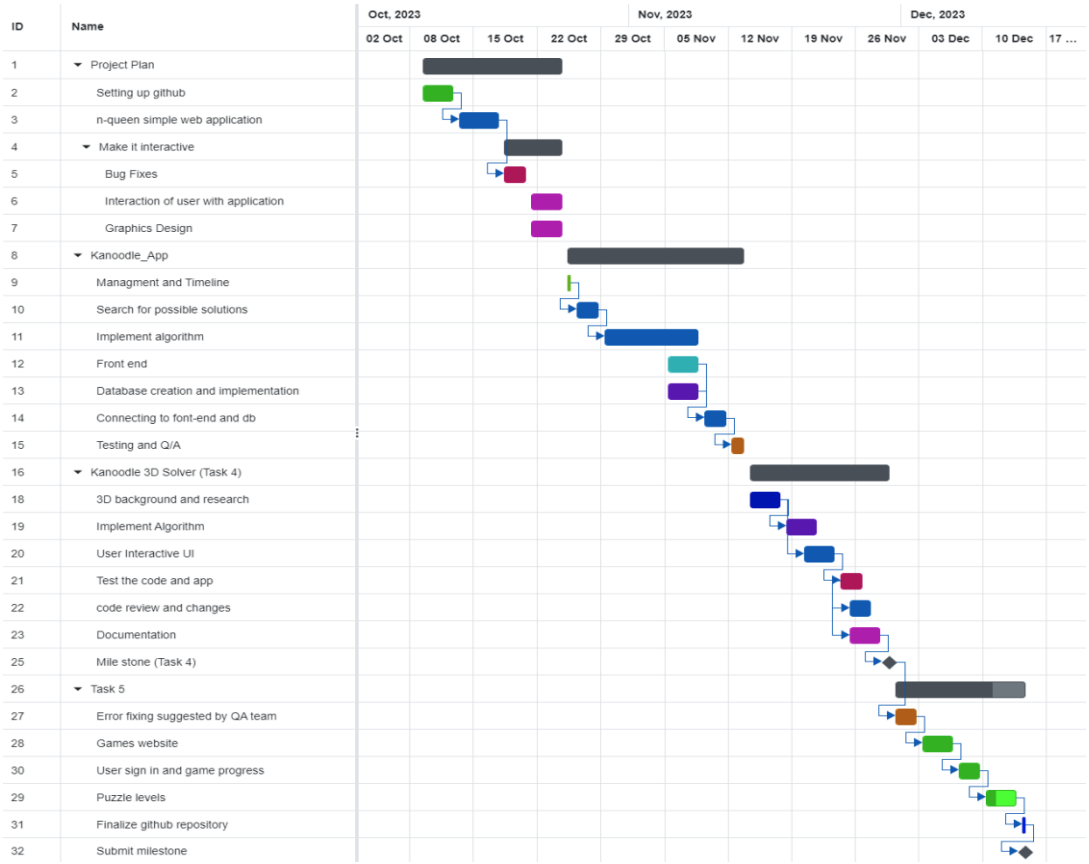

Figure 2: Team Timeline Overview

Our project was centered around creating a puzzle web application that provided an immersive experience. With the client's expectations set high, we embraced the Agile methodology to foster flexibility and responsiveness throughout the development lifecycle. The team worked as a cohesive unit where it primarily focused on the following methodology of delivering tasks to the customer. This practice combined with segmenting of roles enabled everyone to be effective at their best abilities they could offer. During this phase, everyone was assigned as a backup unit for each roles incase any member of the team drops out for some reason. As a result, it developed the members skills in different areas that they never had the chance to catch up to. All commits to the project main branch were disallowed and policies were outlined to use own branches when doing commits by the team leader to ensure there were no conflict among the team members. Each member was assigned to hold accountability of each other including themselves so that the project would go smoothly without any fallbacks. Every milestone was celebrated with a lunch date where we would take a chance to know each other and keep track of each other's personal development to ensure everyone was on the same page.

3 Key Features

3.1 Highlights: MVT Architecture and Collaborative Accountability

One pivotal aspect that significantly bolstered our software engineering process was the adoption of the Model-View-Template (MVT) architecture. This methodology facilitated a granular understanding of our codebase, allowing us to optimize algorithms and enhance program efficiency. By dissecting the flow of program calls, we identified areas for improvement and implemented new logic to streamline our approach.

Moreover, our team embraced a culture of collaborative accountability, utilizing GitHub space to post issues and engage in discussion panels. This not only ensured a transparent and efficient workflow but also fostered healthy competition. Each team member was driven by a collective commitment to excellence and strived to bring out the best in themselves and their peers.

3.2 Lowlights: Django Framework Limitations and the Importance of Flexibility

While our journey was marked by successes, we encountered challenges with our initial reliance on the Django framework to model our entire application. Task 4, which required heavy integration of JavaScript libraries like Three.js for a 3D spatial puzzle, pushed the limitations of Django. We found ourselves deviating from the proposed Model-View-Template pattern, realizing that a strict adherence to the framework impeded our ability to meet the client's expectations.

In hindsight, our team acknowledged the necessity of a more flexible project plan those accounts for potential deviations from the initial architectural choices. Anticipating the limitations of our tools and technologies would have allowed us to allocate resources more effectively and adapt our approach sooner.

4 Challenges and Lessons Learned

Participating in the 8-member group project for web development of puzzles presented a unique set of challenges. Due to diverse skill levels within the team, many of our team members had limited knowledge and were hesitant in using Git as it had a learning curve which led to many merge issues as complexity of our tasks increased. We were finding it very difficult to collaborate and hence we took separate time to understand Git and use it thoroughly in our project which familiarised us with its functionalities.

Initially we started assigning individual tasks of each domain to one member each. We assumed this approach might be most reasonable as then a teammate can contribute with his/her expertise. But soon we realised that there was a huge knowledge gap between teammates which led to major issues in combining each teammate's work. Half of the team possessed limited programming language expertise and were unfamiliar with technologies such as Django, Python, 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 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. 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.

5 Inspirations

Many projects played a major role in influencing our design patterns and development during our endeavor. As we go through several projects that inspired us to do even better, we found some amazing projects that really piqued our intuition. The N-Queens puzzle by [1] demonstrated a use of backtracking algorithm keeping in mind with all the restriction rules of queen on placement. This helped us understand how algorithms generally are integrated into logic for greater computation purposes such as time complexity factor and algorithm efficiency. Likewise, task 3 gave rise to similar inspiration with apps developed by [2] and [3] which guided us with how a working polysphere puzzle had been made using backtracking approach, but this approach provided us with a bit of downfall around time complexity as it would take around 6-7 hours to get full solutions depending on the hardware we were using. This problem paved a way for our team to think differently with a better efficient approach to solve the problem which lead us to a paper published on a similar game theme but a different algorithm [4], which talks about dancing links algorithm which used an exact cover problem approach and helped reduce solving time by the way of reduction. Hence, we then used then attempted to use this algorithm in our task 3 which greatly reduced our solve time to 39 min which was way better than the brute force technique thanks to [5] which helped us have a general idea of how the algorithm functions are assembled to perform. Similarly, task 4 was a bit tedious as it proved much harder than the previous tasks, which was to build a 3D model of the same puzzle as task 3. [6] showed us how using Jscript libraries such as Three.js could help us achieve this masterpiece which kept us awake for many nights to understand the mechanics of 3D space and really develop our programming skills. Lastly, task 5 was free form and did not have any requirements which led to [7] git repository which had a game like UI for the puzzles suitable for kids. This inspired our very own puzzles to be driven as game style with levels introduced and have interactive buttons for people of all ages to play and have immersive experience.

6 Conclusion

In short, our team has found this journey to be both rewarding and challenging. Each task required a different skill set and teamwork from solving the N-Queens puzzle to creating complex 3D environments such as pyramids. 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. Understanding the nature of algorithms and how each of them had their own pros and cons when using on different sets of computational tasks was a great revelation to our thoughts as using algorithms such as the Dancing Links highlighted the significance of strategic decision-making for our team 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.

7 References

[1] Linh. "N-queens problem - visualization." Codepen. https://codepen.io/linhnt1516/pen/VpmJdJ (accessed 2023).

[2] A. Loves. "Kanoodle Challenge Solver." Vercel. https://kanoodlesolver.vercel.app/ (accessed 2023).

[3] A. Ward. "KanoodleJs." GitHub. https://andywarduk.github.io/KanoodleJs/Kanoodle.html (accessed 2023).

[4] M. Harrysson and H. Laestander, "Solving Sudoku efficiently with Dancing Links," 2014.

[5] R. D. Programmer. "Kanoodle Solver using Dancing Links in Java." Reddit. https://gist.github.com/anonymous/75281bc0b5063453126c (accessed 2023).

[6] B. Elder. "Kanoodle Extreme Solver - 3D." GitHub. https://github.com/benelder/kanoodle-3d (accessed 2023).

[7] W. Keeling. "Kanoodle Genius 2D." GitHub. https://github.com/wkeeling/kanoodlegenius2d (accessed 2023).