

ENSE374 — SOFTWARE ENGINEERING MANAGEMENT
THREEWIDE — INTERACTIVE TETRIS LEARNING PLATFORM

Daniil Aleshechkin, Benjamin Hajdukiewicz, DongYun Kim, Tirth Patel

Creative Commons Share & Share Alike (CC BY-SA 4.0)

Fall 2022, University of Regina

This report is best read accompanied by the project GitHub repository. When viewed locally, documentation mentions are hyperlinks to the associated document within the repository.

CONTENTS

1	Introduction	2
2	Exploration	2
2.1	Stakeholders	3
3	Project Execution	4
3.1	Project Kickoff	4
3.2	Project Management Tools	4
3.3	Minimum Viable Products	5
3.4	Design documentation	6
3.5	Project Tracking	6
3.6	Managing Tasks	6
3.7	Encorporating Feedback	6
3.8	MVP 1 Completion	7
3.9	MVP 2 shift	7
4	Reflection and Lessons Learned	8
5	Closing	10

For more visit <https://github.com/teamcrusher/threewide>.

1 INTRODUCTION

Project idea

The idea for Threewide is to create an educational platform for Tetris strategies and techniques where members can interact with an embedded game. Users will be able to catalog strategies or techniques with community feedback where members can share and gain knowledge to familiarize themselves with the game and prepare with proven knowledge of strategy at every level.

Primary Goal

To create an educational Tetris strategy platform with good interactivity.

Accomplishing the goal

Users would be able to read and edit a wiki with static pages to gain and share knowledge about the game of Tetris. These static pages generated via SSR with express JS and stored with markdown in Mongo DB. In order to display the methods properly tetris pieces generated as SVGs.

A React app would let users interact with the knowledge, by playing the strategies presented. The React app would take a board state, piece queue, and solution data as inputs and present them in a way that the user could repeatedly play them to practice. Personalized user settings saved using cookies.

Depending on the topic and user interaction will provide appropriate feedback in the form of a solution check method, with shared metrics to track progress and compare the users ability to previous their games and strategies attempted.

Rationale

Threewide would build a better Tetris community which allows more tetris players to master the game. Sharing knowledge (Skills and techniques) is the key action to enable this. Some communities share this goal with varying levels of effectiveness but fail to provide adequate interactivity to enhance overall game experience. This is due to the lack of user friendly interfaces and accumulation of shared insights. Tetris is more than a game, and the sense of community is strong, but due to the previously mentioned shortcomings requires an initiative such as Threewide.

2 EXPLORATION

To initiate the project, we researched the current state of the Tetris community, what we thought would be the ideal state and how we would start to shift the landscape from the current state to the envisioned ideal state.

Current State

Currently only forum based wikis such as hard drop and four.lol. The only project that lets users practice strategies are twowi.de (which currently is down), jstris maps, zen mode in tetr.io, and fourtris. All of which are unorganized and difficult to link the strategies from the wikis to a practice environment requiring expertise to use them properly.

Ideal State

Players would be able to practice their game effectively without relying on multiple tools. Beginners can learn with simple descriptions of strategies to practice, interacting with the concepts to gain an intuitive understanding.

Action

Develop and deploy a unified platform that contains all tetris knowledge with tools to practice it in one place.

2.1 Stakeholders

A stakeholder register rationalizes the roles, level of interest, support, and power of both external and internal stakeholders building on the initial research in support of the project kickoff.

External Stakeholders

Beginner Tetris players who are passionate about learning want an easy way to learn new strategies and their feedback is essential for our project's success. Incorporating feedback from top players and adapting their knowledge to guide our platform development would strengthen the effectiveness of the project's MVPs. Our audience and users are across the globe — there are no borders in Tetris game and its community.

Internal Stakeholders

The project team itself along with the project sponsor carry the most power and support as they are ultimately responsible of defining the scope of the project and implementing the plan of project execution in alignment with the wants and needs of the external stakeholders.

Engagement

To ensure the engagement of all parties, an engagement plan outlines the responsibilities of each party and what will be monitored and communicated internally and externally.

3 PROJECT EXECUTION

3.1 *Project Kickoff*

With the information gleaned through initial research and reflecting on the project's goals, we developed a project charter, which defined our project stakeholders, initial project milestones, and risks which served as the basis of the business case to further rationalize the project for approval. The business case served to gain alignment on the problem, rationalize the opportunity and technologies implemented, and provide a recommendation of a path forward to the stakeholders. A project scope statement provides the scope of the project's deliverables in the form of Minimum Viable Products (MVPs) to the stakeholders. The communications management plan serves to define the frequency, type, and audience of all stakeholder communications to inform the necessary parties of progress.

These foundational documents allowed for the formulation of project requirements based on the scoping of the MVPs. This determined roles and responsibilities of internal stakeholders, ensuring that each member's skill set would be effectively utilized in accomplishing the project goals by assigning work based on the individual's skill set. Combining this document with the communications management plan served as the basis of the Responsible Accountable Consulted and Informed (RACI) Chart to further solidify the roles and responsibilities of internal stakeholders. The meeting log provided documentation of the team's internal meetings, capturing the majority of large decisions and the path forward after each activity.

Additionally, to manage the development and ensure that proper document and code base reviews were taking place, the team opted to create a quality management plan to align on the branch protections needed, the number of reviews each Pull Request (PR) would need to undergo before merging, and when to bypass the branch protections.

3.2 *Project Management Tools*

GitHub integrates Project Boards into their repositories and these served as a tool to track progress over the course of the project. One Project Board tracks all of the project activities that are directly related to the project's supporting work, such as the creation of documentation, meetings, reviews, activity milestones, and external presentations.

The Dev Board tracks all of the project development work directly related to the building of the application. The decision to separate these two boards was simple, as the two boards represent different disciplines of work. One focuses on the project management tools, while the other focuses on the development, which are somewhat loosely coupled.

The User Story Board captured all of the user stories for the following MVPs.

The project team decided early that the time would be best spent on developing the first MVP and then allocating any additional time to the second MVP, which would be far less ambitious in scope.

3.3 *Minimum Viable Products*

The selection process for the MVPs consisted of re-evaluating the goals of the project after project initiation to determine the suitability of each MVP. Revisiting the primary goal of create a learning platform which helps members — especially beginners, to familiarize themselves with the game and prepare with proven knowledge of strategy at every level.

By providing practical exercises to enhance knowledge though interacting with the user, Threewide will allow players to hone their skills. Threewide would have a intuitive UI/UX for learning and interaction, bringing a more immersive learning experience with each advancing level like Brilliant or chess.com.

MVP 1 — Practice Platform

The initial MVP consisted of a Tetris app built with React. It implemented all the basic functionality of Tetris, with some additional platform specific integrations such as persistent user data to store the user's keybindings, DAS settings, and playing history. The user onboarding process only required a username and password, allowing for fast and easy sign-up for the end user, and ease of development for the project team. The UI/UX closely resembles the prototypes from activity 2, but adopted some student feedback to incorporate more colours and controls that were not accurately implemented in the prototyping phase.

MVP 2 — Educational Platform

In alignment with our primary goal, MVP 2 would add a wiki style strategy guide to help educate Tetris players on the variety of strategies available to succesfully and effieciently clear scenarios. Users would be able to add game board setups associated to the strategies they had written, utilizing the interactive practice tool, the game, to implement the strategies provided. MVP 2 would include strategies for the following topics:

1. Openers
2. T Spin Methods
3. Finesse
4. Perfect clear methods
5. Downstacking methods
6. Kicks

3.4 *Design documentation*

The main design documentation consisted of the MVC architecture overview, the data model, sequence diagram, state diagram, a Tetris flow diagram, and a user journey diagram. These documents not only defined the overall application architecture, but provided key details on the standard types and data components of the application, thereby increasing development efficiency. The data model was particularly useful in onboarding team members to the codebase as it provided all the standard types used, and defined the connections between systems within the application. These documents serve as the basis of which the future application documentation is built, as they accurately reflect the current state of development and can be easily updated as additional MVPs are created in the future.

3.5 *Project Tracking*

Tracking of progress was an essential aspect of the success of the project team. This was facilitated by the use of the project management tools made available by GitHub. They were feature packed with everything needed to effectively monitor progress, from the Project Boards to the repositories Insights.

3.6 *Managing Tasks*

The project team quickly aligned on utilizing the Issue tracking and Pull Request features of GitHub as they easily integrated into the Project Boards without the use of additional tools. This streamlined process allowed the team to not only centralize documentation and development, but the tracking of both. All work was self assigned based on the skills of the team members, with some overlap where members were eager to explore new technologies that they had not used before.

3.7 *Encorporating Feedback*

Feedback from multiple sources including our instructor, peers, and the Tetris community was evaluated and incorporated. Initial instructor feedback was positive regarding the potential of the project, with potential to further explore it in the future beyond the scope of the course, either as a capstone project or a product deployed for public use.

Our peers encouraged us to explore more delightful design in our prototypes by employing the use of colours in our designs, as the initial prototypes focused on the implementation of the interactivity rather than the visual design of the app. Another common theme with this feedback was the presentation of the game states and instructions or strategies on how to play, addressed in MVP 2. Our first MVP implements a colour header and more information regarding how to perform certain moves within the game.

External feedback included some confusion regarding the login page, it is not clear whether a user needs to sign up for an account, or if they required credentials for beta access. This feedback has not yet been incorporated but is captured on the dev board as future work.

Additional feedback included creating games at first on mass, by just having a Tetris game board that allows you to both manipulate the board and queue. Then to create the games, you just play that setup and click record, where all the user actions would be recorded and stored. Actions would then be analyzed to calculate the goal data. From there, they would be another page to modify strategies where a user can search games that haven't been added by name and both add them to the page as well as modify the content.

User Questionnaire

The purpose of the user questionnaire is to determine the usability of the initial Lofi / Hi-fi designs of the UX of Threewide and ensure it flows well by evaluating the ease of executing key activities. The score is not an evaluation of the user, but of the design. The questions posed are meant to encourage exploration of the design and gameplay. The results of the questionnaire are used to improve the design of Threewide, incorporating the feedback gain from the completion of the questionnaire.

3.8 MVP 1 Completion

On November 28th, Threewide was successfully deployed to Vercel and available to the public. This marked a great milestone in the project and

3.9 MVP 2 shift

Initially MVP 2 would focus on the implementation of the wiki pages for the sharing of strategies. This turned out to be a larger task than initially scoped, thus Threewide shifted to more game optimization for MVP 2.

The integrated game functioned, though it was not optimal and some browsers struggled to run it. To resolve this, refactoring the rendering of the game pieces as canvas elements proved to be effective and resolve the initial performance issues, lowering the latency of the UI to <1ms from ~80ms.

A major DAS bug was also resolved in MVP which greatly improved gameplay, as previously a game piece would not always DAS to the game board edge, or sometimes would not DAS after it had been rotated, both are essential moves in Tetris. A refactoring to use WebGL is also planned for MVP 2 for more performance improvements.

4 REFLECTION AND LESSONS LEARNED

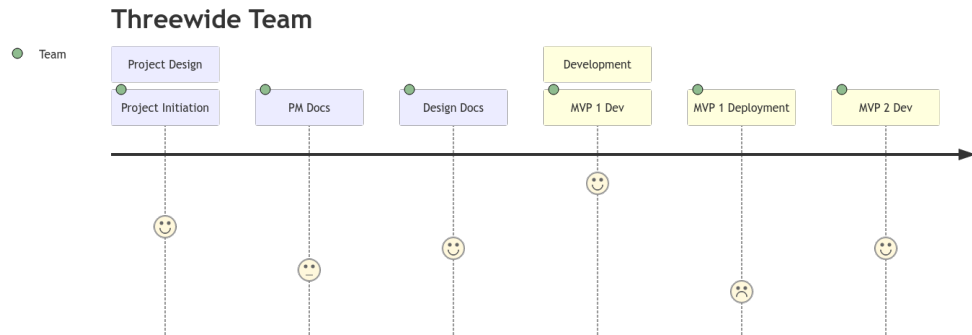
An Honest and Open Journey

Figure 1: Our Journey Map

GitHub provides some excellent tools to glean insights on the members contributions. Though they aren't absolutely reflective of member's contributions, they do help to visualize effort.

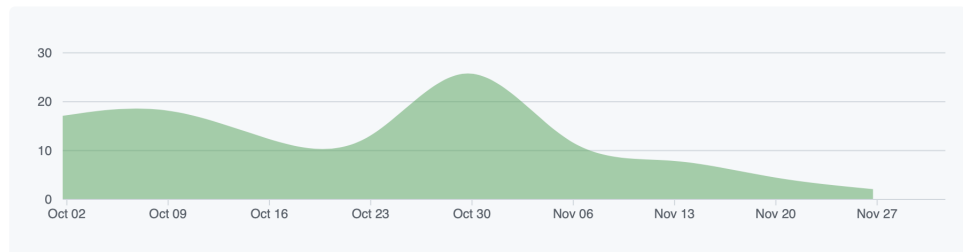


Figure 2: Commits Oct 02 — Nov 27

In Figure 2 looking at commits over the course of the project where the PM documentation and development work was taking place. The distribution is biased towards the work taking place at the end of August, which coincides with the development of MVP 1. The large tail is the commits focusing on the PM documentation. This serves as a visual representation of the level of effort required through each phase of the project, and intuitively feels representative of the team's effort.

Figure 3 shows all additions to the repository, which more accurately reflects the MVP development phase.

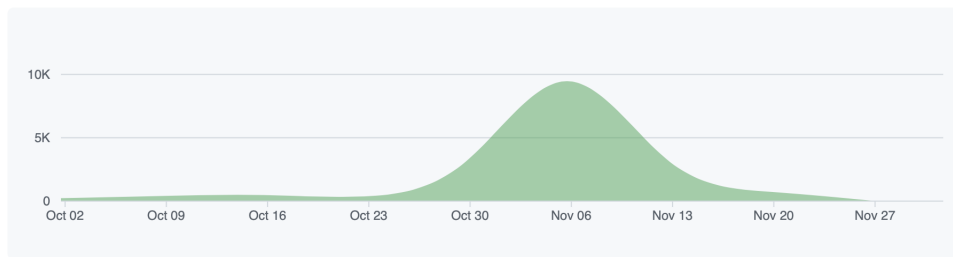


Figure 3: Additions Oct 02 — Nov 27

What went well

Initiating the project was a very exciting time, we had a member with motivation to create an application that they were passion about who was able to motivate and inspire the team. Starting the project off strongly with a well developed goal provided momentum to get through the project management document creation.

At times the project management document creation was a slog, but the team kept the end goal in mind knowing that these documents would serve as a solid foundation for success by providing a strong outline of the project lifecycle to refer to when things inevitably become a little fuzzy.

Design documentation went well, the Tetris flow diagram provided an additional tool for onboarding team members to the code base. The team fit together very well, quickly falling into roles that suited eachother strengths and played to the strenghts of others. Members were eager to support eachother when questions came up or claficiation was needed on certain topics, as the team is diverse in experience, not only of Tetris but web development as well.

In contrast to previous projects in our ENSE classes, the use of a version control system for the project codebase was an asset. It eased deployment of the codebase as well as the review process, allowing for comments on commits and pull requests and fascilitating conversation regarding changes.

The team adhered well to the Agile principles of working software being the primary measure of progress, by implementing features that were well defined by the user stories and aligned with the project's goal.

What went less well

Success does not come without struggles. Self assigning work may not have been the best method of task assignment, with one developer implementing most of the features as shown below. The graphs are not entirely reflective of team contributions as the member who merges PRs into mains gets tracked on the commits, sometimes weighting their merge as additional contributions,

when the member doing the work lost those contributions. Figure 4 provides an breif glimpse of this.

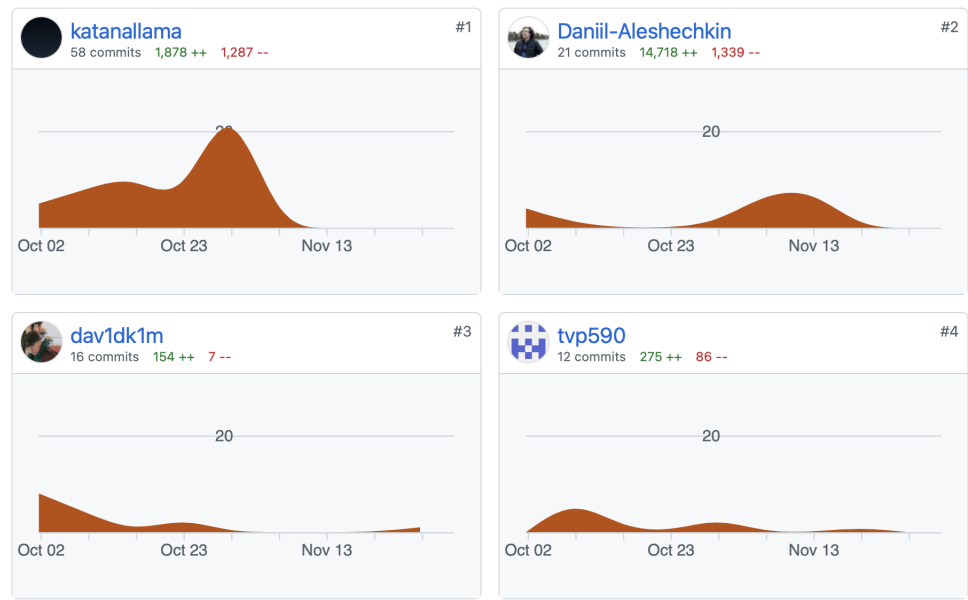


Figure 4: Team Contributions Oct 02 — Nov 27

Future Considerations

Perhaps a more rigid assignment structure would help alleiviate overwhelming one developer. Another consideration would be to maintain the stack that was initially decided upon. Prior to MVP 1, React was the main framework used, then the project shifted to the T3-stack, leading to difficulties in on-boarding the team to the new framework, as it was new to all members and some had more difficulties grokking it than others.

The team tried their best to overcome this change and continue development efforts for MVP 2, but fell short of completing all cards for MVP 2. This leaves plenty of future opportunity to learn the stack implement these features in the future.

5 CLOSING

Overall, the deployment of MVP 1 was a success. The team accomplished the goal, leaving some work for upcomming MVPs. Though the wikis were not implemented, an interactive environment for practicing a few strategies was deployed. A T3-stack app lets users interact with the knowledge, by playing the strategies presented. The T3-stack app takes a board state, piece queue, and solution data as inputs and present them in a way that the user could repeatedly play them to practice. Personalized user settings are saved using cookies.