Table 1: Revision History

Date	**Developer(s)**	**Change**
2/2/2021	Nicholas Lobo	Initial document creation
2/2/2021	Matthew Paulin	Initial document creation
2/2/2021	David Carrie	Initial document creation
2/4/2021	Nicholas Lobo	Section 1,2,7
2/4/2021	Matthew Paulin	Section 3,4,6,8
2/4/2021	David Carrie	Section 5,6,7

SE 3XA3: Development Plan
Tetris Tussle

Nicholas Lobo	Matthew Paulin	David Carrie
lobon3	paulinm	carriedd
400179304	400187147	000661652

1 Team Meeting Plan

Meetings will occur during lab times on Tuesday and if needed Thursday from 3:30 - 4:20 PM and 8:00pm - 9:00pm respectively. Additional meetings can be organized as necessary. They will be held on Microsoft Teams in the Group 2 channel of the Lab2Group3xa3 team. These meetings will occur weekly. Nicholas will act as project manager and start the meeting by stating the deliverables that are expected to be done by the start of the next meeting. Each member of the group will discuss what they have been working on what they would like to complete prior to the next meeting. The group members will also discuss any tasks that they require assistance with.

2 Team Communication Plan

The team will be mainly communicating through Microsoft Teams using text or audio chats. This is beneficial for the TA's as well as Professor Bhokari as they can track the progress of the project and can be aware of interpersonal issues among the team. Team members can message individual members through Microsoft Teams. Git-Lab will be the primary way of managing and adding modifications to the software. Descriptive commit messages will indicate the latest changes made.

3 Team Member Roles

Role	Member	Description
Project Manager	Nicholas Lobo	Responsible for project coordination and communication with the professor and TAs.
Software Architect	Matthew Paulin	Responsible for the high level design and final choice of technology.
Lead Developer	David Carrie	Responsible for overall module design

4 Git Workflow Plan

The main/master branch will contain the latest working version of the software. At the appropriate milestones, it will be updated to reflect all working changes. Team members will not commit directly to the master branch after the initial stages of development, but instead to a development branch. The master branch will also be labeled with a version number that may be altered as enhanced versions are created. Individual bug fixes and features will be completed on their own branches that will be merged to the development branch when finished. Commit messages will accurately and succinctly describe the changes made at each branch. Additionally, a test branch will be used for testing the code from the development branch before merging to the master branch. Every group member will have access to commit to any of the branches, but this workflow will be strictly adhered to.

5 Proof of Concept Demonstration Plan

The most significant risks identified at this stage of development are related to the multiplayer component, the user interface, and gameplay. Therefore, additional measures should be taken to mitigate the adverse outcomes that can result from these areas. We expect the most issues when developing the

multiplayer game mode and the user interface. Implementation of multiplayer will add significant complexity to the design and is an area that our team is inexperienced in. Since this is an important feature, additional precautions should be taken to mitigate risks. Additionally since the main function of the software is a game, the user interface is a top priority and any failures in it will risk the entire project as the user may not be able achieve full game functionality. The final significant risk that we determined is gameplay. If the game does not play, then the product is without a purpose and is useless. Also, if the game is too easy or too hard then it loses its entertainment value. Although testing

is expected to be straightforward, additional effort will be spent on ensuring the multiplayer and the user interface is functioning properly. In general, we

expect standard unit testing for our system, with extra time spent on testing UI, functionality, and performance. There are no difficult libraries to install

and portability will not be a concern for the project. The software will be designed to play in modern web browsers that can run HTML5 and is expected to be playable on multiple devices. To mitigate and show how we will overcome

the risk with multiplayer, we will display that two instances of our game can communicate during the proof of concept demonstration. This will ensure that we have a foundation for building upon and that the multiplayer feature is attainable. We will also have a functional user interface to ensure the game can be played enjoyably, along with the game being playable in a demonstrative state.

6 Technology

The project will be developed using HTML5, CSS3, JavaScript, and the P5.js JavaScript library. Additional JavaScript libraries can be installed and used as needed by using the Node Package Manager. The page markup and styling will be completed with HTML5 and CSS, while the game itself will be created with JavaScript and P5.js. These technologies were chosen for their robust documentation as well as the ease of creating a game.

Microsoft Visual Studio Code (VSCode) will be used as a code editor. It contains features that assist in formatting and debugging, IntelliSense for auto completion, a built in terminal, and embedded Git support. The community made extensions add on to this functionality, one of which enables collaboration similarly to Google Docs, allowing for effective pair coding.

Doxygen will be used for automatically generating code documentation. Doxygen is the de facto standard tool for generating documentation from annotated JavaScript.We will use it to generate an on-line documentation browser in HTML and an off-line reference manual (in LaTeX) from the documented source files. Since the documentation is generated directly from the source code, it is very easy to keep the documentation up to date and consistent. Code structure can also be extracted for navigation and even visualized if needed.

For the testing framework, we will be using Jest. Jest is a JavaScript Testing Framework that is focused on simplicity.

7 Coding Style

The Prettier formatter will be used to enforce the JavaScript coding style that will be used for this project. The details for this guide can be found here.

8 Project Schedule

Our Gantt Chart can be found here.

9 Project Review

Will be done in revision 1.