Table 1: Revision History

Date	**Developer(s)**	**Change**
Febuary 5th 2020	Abdallah Taha	Added sections 1-3
Febuary 5th 2020	Ali Tabar	Added intro blurb, section 4, revised grammar of other sections
Febuary 5th 2020	Andrew Carvalino	Added section 7 and MacIDs
April 11th 2021	Ali Tabar	Revised sections 5 and 6 for last revision.
...	...	...

SE 3XA3: Development Plan
Title of Project

Team 115, AAA Solutions
Abdallah Taha, tahaa8
Ali Tabar, sahraeia
Andrew Carvalino, carvalia

This document will provide detail on the development plan of AAA Solution's Tetris game. It includes a team meeting plan, plan of communication, list of the different members' roles, the workflow plan for working with Git, proof of concept demonstration plan, list of technologies being used, coding style, and a pointer to the Gantt chart of the project schedule.

1 Team Meeting Plan

AAA Solutions plans to meet through Discord voice chat once a week, on Fridays at 3:30 pm Eastern Standard Time, for at least 30 minutes. However, on weeks in which there is a deliverable due we will also meet Tuesdays at 4:30 pm Eastern Standard Time, for a period of at least 30 minutes. During these weekly meetings, Andrew Carvalino will be the scribe and is in charge of keeping a record of the meeting. The meeting chair will alternate between Abdallah Taha and Ali Tabar. Each week, we will alternate responsibility of writing down a statement of decisons made for the upcoming week.

2 Team Communication Plan

AAA Solutions will use primarily Discord for our main source of communication. When individually working on a deliverable, it will be common practice to include commit messages through git, to inform other team members on the progress that has been achieved.

3 Team Member Roles

Abdallah Taha: Team Leader, Full Stack Developer
Ali Tabar: Project Coordinator, Full Stack Developer
Andrew Carvalino: Scribe, Front-End Developer

4 Git Workflow Plan

Team members will individually work on their sections, though with constant dialogue and feedback between them at all times during the process. Once each team member is done, they will commit and push their changes, letting the other team members know. The other team members will first pull from the repo, before committing and pushing their own changes, too. A git pull should always be done before a member decides to commit their changes to the repo.
Major changes to the code should be kept track of, such as drastic restructures or other large-scale edits. Whether a change should be deemed as a major one will be discussed by all team members. Major changes will be kept track of by making a branch to the repo, with git branch and git checkout.

5 Proof of Concept Demonstration Plan

We plan on putting together an initial prototype without functionality. We will develop a simple UI to demonstrate the presentation of our game. In the future, we will develop the game mechanics to work within terminal, to prove that the game is functional.
Some risks we are facing in our implementation are to do with learning how to connect the front-end and back-end - this is in regards to ”connecting” the HTML file to the JavaScript file, and which components should be in which file. We will try and overcome these risks by discussing and setting up group learning sessions to look over JavaScript-HTML tutorials online and write base test code together. These sessions will be staggered in a way that will not be overwhelming, but time-efficient, ensuring that the majority of concepts can be learned well before any major due date comes up for a prototype.
For testing functional requirements, we will rely on mainly user tests – this is due to the nature of our program. Since it's a game that only allows user inputs that it can recognize (ex: clicking of a button, pressing keys to control blocks), we can recreate scenarios we want to test in the game, and note the outcomes the program gives. In addition, we can test certain game scenarios by programming an environment with only certain aspects of our game (ex: shapes are allowed to rotate, but don't ever fall) and automate these parts with test cases. We can use Jest for automating these.
For testing nonfunctional requirements, such as testing if the game is more easily navigable and looks better than the original implementation, we will be conducting test groups, with surveys for people to fill out with their opinions afterwards.

6 Technology

For our front-end, we will be using React, which is a JavaScript framework. Our back-end will be programmed using Python. All members of AAA Solutions will be using VSCode as the IDE for writing the code. Documentation will be done manually, using LaTeX for writing up PDF documents that will log different aspects of our program, such as the Module Interface Specification or the Module Guide.

7 Coding Style

For development in JavaScript, the following styling guide will be used:
https://github.com/airbnb/javascript/tree/master/react
For development in Python, the following styling guide will be used:
https://www.python.org/dev/peps/pep-0008/

8 Project Schedule

The Gantt chart of our project schedule is located in this repo, at 3xa3-L01-group-15/ProjectSchedule/group115.gan.

9 Project Review