Table 1: Revision History

Date	**Developer(s)**	**Change**
Date1	Name(s)	Description of changes
Date2	Name(s)	Description of changes
...	...	...

SE 3XA3: Development Plan
Title of Project

Team #, Team Name
Student 1 name and macid
Student 2 name and macid
Student 3 name and macid

Put your introductory blurb here.

1 Team Meeting Plan

Team meetings will occur once per week for about thirty minutes after our first lab of the week. Our first lab is on Wednesdays from 8:30 to 10:20 in ITB , and we will have our meeting at the caf in ETB immediately after the lab. This way all of us can easily attend the meeting since we will all have met during the lab. We will have a team chair to keep the meeting on topic, a scribe who will keep record of the meeting with a meeting agenda, and the last person will keep track of how well the implementation and program requirements fit together.

The meeting agenda will divide the meeting into three equal parts - reviewing the progress for the current week's deliverables, discussing any problems/changes with the implementation and documents, and dividing the work to be done for the next week's deliverables. This agenda is general and topics are subject to change over time.

2 Team Communication Plan

3 Team Member Roles

Our team has agreed on rotating team roles per meeting. One person will lead the meeting while the other person will note down all the goals discussed in the meeting. The third person will be in charge of making sure the project is going in the same direction as the program requirements. Our team is very well balanced because everyone has intermediate to advanced knowledge of LaTeX, Java and Git.

4 Git Workflow Plan

Our team will be developing our program using the Centralized Workflow plan, keeping a central repository to serve as a single point-of-entry for any and all changes made to the project (changes will be committed to the default *master* branch). Since we are all relatively new to using Git, this will help us easily keep track of each other's changes to our project files. We will use tags to keep track of important deliverables and personal milestones for the project, such as when major portions of our program may be implemented to work. Tracking these commits will ensure that we can fallback should we come across issues with our game, and that we can have separate releases if we are able to implement extra features.

5 Proof of Concept Demonstration Plan

6 Technology

Namcap will be programmed in Java and will be organized as an OOP project. The UI/Graphics will be implemented by utilizing the Java Swing package. All the programming will be done using the Eclipse IDE. To test the robustness of the program, we will use Java's Robot class to program the player to make all the possible moves that can be made at any point of the game. This will test the functionality of all the objects in the game (barriers, points, enemies, end game, etc.). Lastly, all the documentation will be created using LaTeX in order to maintain the same formatting and style.

7 Coding Style

8 Project Schedule

Provide a pointer to your Gantt Chart.

9 Project Review