Table 1: Revision History

Date	**Developer(s)**	**Change**
23/09/2016	P.S., V.R., V.G.	Initial Drafts of Communication, Roles
27/09/2016	P.S., V.R., V.G.	Initial Drafts of Meeting Plan, Technology, Coding Style
29/09/2016	P.S., V.R., V.G.	Final Drafts of all sections
30/09/2016	P.S., V.R., V.G.	Final Draft of Revision 0
03/12/2016	P.S., V.R., V.G.	Revision 1
...	...	...

SE 3XA3: Development Plan
PineSweeper

Team 07, PineApple
Prince Sandhu; sandhps2
Varun Rathore; rathorvs
Vishesh Gulatee; gulatev

September 30, 2016

1 Abstract

Using the intrinsic software engineering design principles, the fundamental mission of *PineApple* is to implement *PineSweeper*, a software gaming application that not only individuals can utilize to entertain themselves, but also challenge their mental faculties. *PineSweeper* will be a replication of the traditional game of MineSweeper, which can be found installed on almost every computer. The following document comprises of *PineApple*'s plan to ensure the successful development of the application.

2 Team Meeting Plan

The meetings are to be performed either in the lab, a library on campus of *McMaster University* or via *Google Hangouts*. During the lab, the meetings are to take place within the allotted two hours, as it also provides *PineApple* the time to work on the program and documentation together. The target frequency is four to five meetings per week, with two meetings held in the lab and the rest outside of lab time. Regarding the roles, the chair and scribe positions are to alternate amongst the team members with each meeting. The appointed chair is assigned the task of leading the meeting, whereas the scribe is to record the decisions made during the meeting. Lastly, Dr. Spencer Smith and the teaching assistant, Christopher McDonald, are the communicators; any issues that arise while the project is in progress can be addressed to them.

3 Team Communication Plan

Communication outside of campus and the meetings will be accomplished using various online tools. To schedule meetings, discuss project progress, and

minor issues, *PineApple* will be using a group chat via Facebook Messenger. Queries and issues specific to the codebase of the application and its feature will most likely be raised, discussed and resolved using Gitlab's issues tracking functionality, as well as using pull requests when implementing feature branches. In addition, every member has exchanged their contact information, including their cell phone numbers, so they can still be contacted if they do not have access to the internet.

4 Team Member Roles

Upon collective agreement, there is no group leader of *PineApple*, instead each member is responsible for supervising certain aspects of the design procedure. As such, the team member roles are as follows: Prince Sandhu plays the role of the programmer, covers the documentation, and is the illustrator. Varun Singh Rathore is also a programmer, and plays a significant role in testing. Vishesh Gulatee plays the role of the programmer, and specializes in Git. If there is a conflict between two team members, the third member is to take a neutral stance and serve as a mediator.

5 Git Workflow Plan

Git will be implemented in the development workspace using Feature Branch Workflow, which is a variation of the Centralized Workflow system. The centralized workflow allows each team member/developer to clone a copy of the entire *PineSweeper* project from the central repository or the default development branch called "master". The local copy is termed the local repository and the additions and changes made to this repository will be stored locally. These changes will be published to the official project in the central repository via git commands such as commit and push, updating master branch. The changes/additions can be reviewed or experimented on by other developers of *PineApple* by "fetching" the updated central commits into their local repositories. Hence, this system allows the team to manage conflicts between the central and local repositories by rebasing (adding changes on top of what everyone else has done) by fetching the project from master and then updating the central repository. Furthermore if local changes directly conflict with upstream commits, Git provides the developers with the ability to manually resolve conflicts.

Using dedicated branches (other than main) for all feature development in the workflow allows the team to model the development workspace after the design architecture used for the *PineSweeper* project (Model, View and Controller Architecture). This encapsulation allows multiple developers to collaborate on a particular feature, all the while ensuring that codebase in master branch isn?t disturbed or contains broken code. Using feature branches also allows the team to streamline communication, constricting the focus of the discussion to a par-

ticular feature and problem, at any given time. For more general issues and focuses, GitLab's issue tracking system will be used. Throughout the development and documentation phase, progress will be marked by the use of tags. This allows the team to organize key commits for future references.

6 Proof of Concept Demonstration Plan

During the proof of concept demonstration, *PineApple* expects to demonstrate a simplistic user interface of the *PineSweeper* game board, which must be able to reveal what is underneath the covered tile when the tile is clicked upon by the user. It must also be able to generate *PineMines* and numbers corresponding to the number of mines bordering each cell. The proof of concept must be executable in the Windows and Mac operating systems.

Regarding the risks, the *PineSweeper*'s properties and interface will be examined in order to look at any uncertainties. Some of the aspects of the game that will be examined include the clicking of the buttons to reveal what is beneath it, and the number of mines in the game. A significant risk that *PineSweeper* may face in the process includes the unit testing of the Model class for gameplay validation. On the other hand, any required libraries are not difficult to install, portability is not a significant concern, and the game is not to be run on a server, so concurrent players and server capacity is not a risk.

7 Technology

The programming language to be used in the development and implementation of this project is Java, and will be accomplished using the Eclipse Integrated Development Environment. Generating documents involves a two-step process; first rough drafts and notes using Google Documents are constructed, followed by generating official versions of the documents in *LaTeX*. The documents are linearly committed to directories in master branch in Git. The Java code is to be documented using JavaDocs and generated using *Doxygen*.

Testing of the application will involve degrees of internal testing (α-testing) and external testing (β-testing), based on the software development phase. This includes conducting both manual testing and test automation.

The manual testing phase will be conducted in the early stages of development and implementation, since the program will be undergoing various changes and updates. It will consist of front-end testing and back-end testing, that will be implemented by a combination of developers of the codebase (including both programmers who worked on the feature and programmers who did not). The next phase of manual testing is to include survey testing (or open β-testing) which will be conducted using unbiased potential users of the application. This

type of testing necessary as it allows for usability testing or testing the human experience and integrating consumer feedback to the validation process of the software development procedure.

Automation testing will include modular testing of each integral feature (especially Model), black box unit testing via /textitJUnit and black box automated integrated testing (testing the application as a whole), as part of closed α and β phases.

8 Coding Style

Google's Java Style Guide is to be utilized for this project. In addition, *PineSweeper* is to be implemented using the Model, View, and Controller (MVC) architectural framework.

9 Project Schedule

Link to *PineApple*'s Gantt Chart

10 Project Review

PineSweeper has been a successful project over the past few months for *PineApple*, who achieved all of the goals detailed before the commencement of the project with a great degree of success. Not only did *PineApple* manage to successfully replicate the open source project from scratch and make numerous improvements, it managed to do the aforementioned by adhering to the software design process, which included providing the appropriate formal documentation. The cornerstone project underscored the iterative characteristics of the software development process. Throughout the course of the project, the team maintained exceptional communication and learned valuable project and time management skills.

One aspect of the project that did not go exactly as planned was conforming to the project's gantt chart. The gantt chart, which is a significant project time management tool, serves as a guide for each team member when direction is needed. Even though the team was ambitious in the gantt chart, it was somewhat difficult meeting most of the internal deadlines, as the deadlines of assignments from our five other courses played a significant role. Nevertheless, *PineSweeper* remained relatively organized, as the team was able to meet all deadlines with ease.

Regarding the in-person team meetings, conforming to the schedule was challenging at times. The aforementioned is due to one of the team member being a commuting student. Nevertheless, if certain meetings were required, they would be conducted on *Google Hangouts or Skype*. *PineApple* was satisfied by the team roles and the team communication and would not change the plan if it were to embark on another project.