Table 1: Revision History

Date	Developer(s)	Change
2016-09-30	Prajvin Jalan	Update Team Meeting Plan and Git Workflow Plan sections
2016-09-30	Vatsal Shukla	Update Team Member Roles and Technology sections
2016-09-30	Prajvin Jalan	Update Revision History and Header
2016-09-30	Baltej Toor	Update of Team Communication and Proof of Concept Demo. plans
2016-09-30	Prajvin Jalan	Update Coding Style section
2016-09-30	Baltej Toor	Edit to the Team Meeting Plan, Team Member Roles, and Technology sections
2016-09-30	Vatsal Shukla	Addition of Gantt project pointer

SE 3XA3: Development Plan NAMCAP

Team 2, VPB Game Studio
Prajvin Jalan (jalanp)
Vatsal Shukla (shuklv2)
Baltej Toor (toorbs)

2016-09-30

Feel free to call this section *Abstract* - CM

This document is a Software Development Plan for Namcap, VPB Game Studio's redevelopment of the classic arcade game Pacman. It contains all the details of the team members' roles, their plans for meetings and communication, as well as general details of the program's implementation.

1 Team Meeting Plan

Avoid terms that could be interpreted as slang, see *Caf* - CM

Team meetings will occur once per week for about thirty minutes at 2:30 on Tuesdays. 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 to discuss expectations before the week's labs. Team members will be able to attend consistently due to all other course work ending just prior to the meeting time. 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 someone to 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 (if not referring to the Gantt schedule). This agenda is general and topics are subject to change over time.

2 Team Communication Plan

External communication will primarily be through the online tool, Google Hangouts. The team will utilize the mobile and desktop versions to make contact in real-time in regards to documentation, technical issues, and the discussion of upcoming deliverables.

Secondary communication will be through email. This form of communication will be reserved specifically for instances where research material/external resources are being looked into for use in the redevelopment process.

3 Team Member Roles

Our team has agreed on rotating team roles per meeting. One person will lead the meeting while another person will act as the scribe and note down all te discussion points and decisions made during 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 well balanced due to everyone having 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 the implementation, and that we can have separate releases if we are able to implement extra features.

5 Proof of Concept Demonstration Plan

In terms of the redevelopment, there are difficulties involved with recreating the mechanics of the game. Developing actions in regards to movement, scoring, and collisions between the main character and various entities in the game will require the most work. Since the original project (JavaPacman) has poor documentation and code organization, the redevelopment will have to focus on the correction and modularization of the code and file structures while redeveloping the core game mechanics.

Since the application will be written in Java, GUI functionality and testing will rely on built-in libraries and classes. Testing specifically will involve automation through certain built-in classes to emulate user input. Due to Java's platform-independence the majority of issues, again, stems from the redevelopment of the mechanics of the core gameplay.

In order to demonstrate that the possible risks involved with game mechanic redevelopment can be overcome, the project will, again, focus on the correction and modularization of the code and file structures along with a proof of concept demonstration for the mechanics. Specifically, the demo will:

- Demo movement capabilities through a square track (showing turning and doubling-back character direction)

- Demo scoring through a display which increments score on character collision with score-related entities

- Demo collision with enemy character released from a holding area along the square track to show appropriate response

6 Technology

Namcap will be programmed in Java and will be organized as an OOP project. The GUI elements will be implemented by utilizing the Java Swing package. Programming will be done using the Eclipse IDE along with supplemental use of basic text editors such as Sublime Text to quickly adjust the implementation. 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

Do provide links if necessary - CM

We will use Google's Java Style Guide as our coding standard for Namcap. Currently we plan to be consistent with all the rules set by this standard except for block indentation; we will be using tabs for indents (or +4 spaces) to improve readability of the code, as opposed to the +2 spaces suggested by the guide. We will make sure to keep track of any other changes we make to our coding style over time, but overall we will remain consistent with all guidelines we set for the code. To enforce the rules, when we cover implementation details during team meetings, we will ensure that the implementation follows the coding standard.

8 Project Schedule

Be sure to include some text to emphasize it is a link - CM

Namcap Gantt Project

9 Project Review