Table 1: Revision History

Date	**Developer(s)**	**Change**
23/09/2016	Or	Create outline .tex

SE 3XA3: Development Plan
Title of Project

Group #6, Team Rogue++

Ian Prins	prinsij
Mikhail Andrenkov	andrem5
Or Almog	almogo

Friday, September 30, 2016

Put your introductory blurb here.

1 Team Meeting Plan

Meetings will be held once weekly, in Thode library at 3:30pm on Wednesday. These weekly meetings will be chaired by the team leader. The team leader will be responsible for developing a rough meeting agenda and ensuring the meeting follows the agenda. This agenda will no be posted but will be briefly outlined to the meeting participants. Full minutes will not be recorded, but the meeting scribe will record the outcomes of the meeting. Any such outcomes will be posted to the team git repository. If a team member cannot make the meeting, a brief summary of the meeting and links to its outcomes will be posted in the team slack channel by the meeting scribe. Any changes to the meeting format, location, or time will be posted to the team slack channel by the meeting chair.

2 Team Communication Plan

3 Team Member Roles

Mikhail will be the team leader. The team leader is responsible for chairing the meetings, allocating work to the team members, and ensuring that all team members are up-to-date about the project status and deadlines. The other team members will alternate fulfilling the role of meeting scribe. Outside of meetings, various team members will fill the role of experts in the project technology. Mikhail will git LaTeX expert, Ori testing/Git expert, and Ian C++/libtcod expert. Expert roles don't constitute work allocations, but rather an indication of who should make sure to be up-to-date on portions of the project and who questions should be directed to.

4 Git Workflow Plan

5 Proof of Concept Demonstration Plan

To demonstrate the feasibity of the project, a proof of concept will be developed. The PoC will demonstrate the following features:

- Basic dungeon generation, including rooms, corridors, and placement of gold, items, monsters, and traps
- Line of sight and pathfinding implementation
- Non-functional items and traps implemented
- Minimum viable monster AI
- Basic movement and very simple environmental interaction (picking up items, basic combat)

It is unlikely that a straight-forward implementation of the above features will prove unusually difficult. It may however be somewhat difficult to implement these features in sufficiently extensible manner that they can be reused without readjustment. The reverse-engineering of the algorithms for various features from the original source is likely to prove difficult. In addition integrating tests into the implementation may prove difficult as no team member has much experience with testing or testing frameworks. At this time the project is planned to only be developed for Linux systems, so portability is not expected to be an issue. The required dependencies have already been installed, with a test application running, so that should remain a non-issue going forward.

6 Technology

7 Coding Style

8 Project Schedule

Provide a pointer to your Gantt Chart.