

Table 1: Revision History

Date	Developer(s)	Change
09/23/2016	Or	Create outline .tex
09/25/2016	Mikhail	Added Communication Plan and Technology
...

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 not 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

Team communications will be distributed across several social platforms. This way, conversations can be grouped together according to formality, visibility, and accessibility. In particular, Slack will be used to exchange informal messages in order to convey meeting times and briefly discuss topics. Team members will be required to check this service at least once per day in order to facilitate quicker response times and encourage more open communication. Next, the GitLab ITS will serve as an official means of tracking bugs, reporting issues, and announcing any other code correction request. All other communications will be performed in person during the lab sessions as well as the weekly

group meeting. Currently, there is no plan to include any other communication streams as this will most likely dilute the conversations to a point where accessibility and traceability is sacrificed.

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 feasibility 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)

If there are no issues implementing the above features, it is assumed there are no fundamental flaws with the requirements or architectural design of the project. There are major features of the project that are touched by the above (advanced item manipulation, traps, hidden passageways, and complex monsters) but these have been left out of the PoC as too ambitious. So long as the underlying code is well architected, these more complex features should flow out of the PoC foundation.

It is unlikely that a straight-forward implementation of the PoC 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

The technology behind the Rogue Reborn project was selected to facilitate a productive development process and an effective user experience. At the very core, C++ will serve as the primary programming language for this application. Aside from the obvious performance and community benefits, C++ also dominates the gaming industry as the language of choice for many studios. Another advantage of C++ is that it contains a binding for the *libtcod* graphics library. This library offers a simple interface to draw ASCII-style art and collect user input — perfect for implementing a terminal-style game. For these reasons, the Rogue++ team decided that developing a C++ project would yield the best experience from both a practical and technical perspective.

Aside from the functional code itself, there are several other considerations that constitute the technology environment. With regards to the IDE, every team member is free to use a text editor of their choice. Imposing a constraint on the development environment may result in unnecessary complications and will undoubtedly interfere with the personal preferences of individual team. Next, to gain confidence in the correctness and versatility of the code, the Boost Test framework will be at the forefront of the Rogue Reborn unit testing effort. This library was selected for its superb documentation, simple approach to test creation, and robust assertion support. Lastly, all design documentation will be generated using LaTeX, while the documentation of the source code will be delegated to the Doxygen tool.

7 Coding Style

8 Project Schedule

Provide a pointer to your Gantt Chart.