Table 1: Revision History

Date	**Developer(s)**	**Change**
Sept 28, 2016	All	Created the Development Plan - Rev 0
Oct 5, 2016	Susan Yuen	Edited Development Plan to reflect decision to use Unreal Engine 4.
Oct 21, 2016	Susan Yuen	Edited Development Plan to reflect decision to use XNA Game Studio and Visual Studio for development.

Blaze Brigade

- Development Plan -

SFWR ENG 3XA3 - Section L02
007 (Group 7)

Jeremy Klotz - klotzjj
Asad Mansoor - mansoa2
Thien Trandinh - trandit
Susan Yuen - yuens2

October 21, 2016

1 Team Meeting Plan

Week	Meeting Dates	Meeting Agenda
Sept 12 - Sept 18	Sept 14: 10:30am-12:30pm Sept 16: 8:30am-10:30pm	Formed the Blaze Brigade team.
Sept 19 - Sept 25	Sept 21: 10:30am-12:30pm Sept 23: 8:30am-10:30pm	Decided on the project and wrote the problem statement.
Sept 26 - Oct 2	Sept 28: 10:30am-12:30pm Sept 30: 8:30am-10:30pm	Created the development plan.
Oct 3 - Oct 9	Oct 5: 10:30am-12:30pm Oct 7: 8:30am-10:30pm	Created the Software Requirements Specification.
Oct 10 - Oct 16	Oct 12: 10:30am-12:30pm Oct 14: 8:30am-10:30pm	Began the Proof of Concept.
Oct 17 - Oct 23	Oct 19: 10:30am-12:30pm Oct 21: 8:30am-10:30pm	Finalized the Proof of Concept and began the Test Plan.
Oct 24 - Oct 30	Oct 26: 10:30am-12:30pm Oct 28: 8:30am-10:30pm	
Oct 31 - Nov 6	Nov 2: 10:30am-12:30pm Nov 4: 8:30am-10:30pm	
Nov 7 - Nov 13	Nov 9: 10:30am-12:30pm Nov 11: 8:30am-10:30pm	
Nov 14 - Nov 20	Nov 16: 10:30am-12:30pm Nov 18: 8:30am-10:30pm	
Nov 21 - Nov 27	Nov 23: 10:30am-12:30pm Nov 25: 8:30am-10:30pm	
Nov 28 - Dec 4	Nov 30: 10:30am-12:30pm Dec 2: 8:30am-10:30pm	
Dec 4 - Dec 7	Dec 7: 10:30am-12:30pm	

2 Team Communication Plan

The team will use Skype for communication outside of the agreed upon meeting times. These lines of communication are available when team members require assistance with their assigned work or require input from other team members on a topic of question. The team will also be using Slack to organize announcements regarding project development and deliverable deadlines.

3 Team Member Roles

- **Jeremy Klotz**: Algorithms Specialist, Developer
- **Asad Mansoor**: Tester, Developer
- **Thien Trandinh**: Gameplay Mechanic, Developer
- **Susan Yuen**: Git Master, Product Architect, Developer

4 Git Workflow Plan

After considering the different types of workflows, we concluded that **centralized workflow** best fits the requirements of this project. This is due to the fact that the project is relatively small - only spanning 12 weeks, and the team will see the project through from start to finish over this time period. As such, a release branch separate from a development branch is not necessary. In addition, team members will be working on aspects of the game pertaining to the same feature or features that rely on each other, so creating any additional feature branches are also unnecessary. Due to these reasons, feature-branch and gitflow are excluded. As a result, we decided on maintaining only one branch, and are thus implementing the centralized workflow for our project. Labels will be used to label any commits containing documents that are graded.

5 Proof of Concept Demonstration Plan

The proof of concept demonstration shall consist of the layout of the software architecture, including the skeleton of the majority of classes, functions, and implementation of the Model-View-Controller software design. The program shall have a working grid implemented, as well click detection and mouse functionality. The game shall also have one unit, which the player shall be able to move by first selecting the unit, then selecting another position on the grid to move the unit to. The player shall also be able to select and deselect the unit by clicking on the unit repeatedly.

Will a part of the implementation be difficult?

There is no significant risk other than implementing all minor details within the given time constraint.

Will testing be difficult?

Testing will not be difficult as the team members have experience with unit testing in frameworks such as JUnit. Although our project will be coded in C#, the unit tests will share similar concepts and ideas to our previous experience with JUnit. As such, our automated unit tests will be able to cover whitebox testing of single functions and state variables.

Is a required library difficult to install?

No. The programs that will be used to develop, which include Visual Studio 2015 and XNA Game Studio, are available for free download to students in McMaster's Software and Computing Department. The downloads are available online through McMaster's CAS department.

Will portability be a concern?

Portability will be a concern as the platform that the game supports only includes Windows, due to the restriction of the software the game is developed with. Visual Studio and XNA Game Studio are supported by Microsoft, therefore the game will only be able to support Windows devices.

6 Technology

- **Programming Language**: C#
- **IDE**: Visual Studio 2015, XNA Game Studio
- **Testing Framework**: Visual Studio Unit Testing Framework
- **Documentation**: LaTeX
- **Other**: Git, Photoshop

7 Coding Style

The coding style for the project will follow Microsoft's C# Programming Guide (https://msdn.microsoft.com/en-us/library/ff926074.aspx).

8 Project Schedule

Please refer to the .gan file for access to the project Gantt Chart.

9 Project Review (for Revision 1)