



Blaze Brigade

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

November 30, 2016

Team Members

Jeremy Klotz - Algorithms Specialist, Developer

Asad Mansoor - Scribe, Developer, Tester

Thien Trandinh - GUI Developer, Gameplay Mechanic, Programmer

Susan Yuen - Git Master, Product Architect, Developer, Designer

Outline

- Team Members
- Purpose
- Scope
- Background
- Development Methodology
- Project Management
- Design
- Demo
- Maintainability
- Robustness
- Feedback
- Next Steps
- Conclusion
- Questions? Comments?



Purpose

- Tribute to popular game genre: *tactical simulation role-playing game*
- Provide a form of entertainment through strategic challenges
- Based on open-source freeware Tactics Heroes
- Re-implementation to improve user experience:
 - Integration of new functionalities
 - Enhanced graphical display

Scope

- Blaze Brigade is a turn-based multiplayer game
- Game state is governed by constant interaction of moving and attacking units
- Objective to win by eliminating all of the enemy units
- Few functional requirements
 - The game shall be turn-based
 - One side will be victorious when the other side has no playable units left
- Few non-functional requirements
 - The game shall be playable by persons who can operate a computer and pointing device (mouse or touchpad)
 - The game shall be playable on computers running Windows operating system, with a screen size greater than 1600 x 960

Background

- Blaze Brigade to refurbish Tactics Heroes, an open source freeware
- Integrating new features that were missing in Tactics Heroes to enhance experience
- Largely inspired by Japanese game series Fire Emblem published by Nintendo
- Fire Emblem revolutionized the genre by incorporating RPG Element

Blue Team vs Red Team

Warrior

Mage

Archer



Archer

Mage

Warrior



Development Methodology

Technology

- Visual Studio
- XNA Game Studio

Version Control System

- Git

Git Repository Manager

- GitLab
 - 259 Commits
 - 12 Branches
 - 5 Tags

Project Management

Project Time and Resource Allocation

- Gantt Chart

Decomposed Task Allocation

- GitLab Issues
 - Created and closed 69 issues
 - 5 milestones being 100% completed

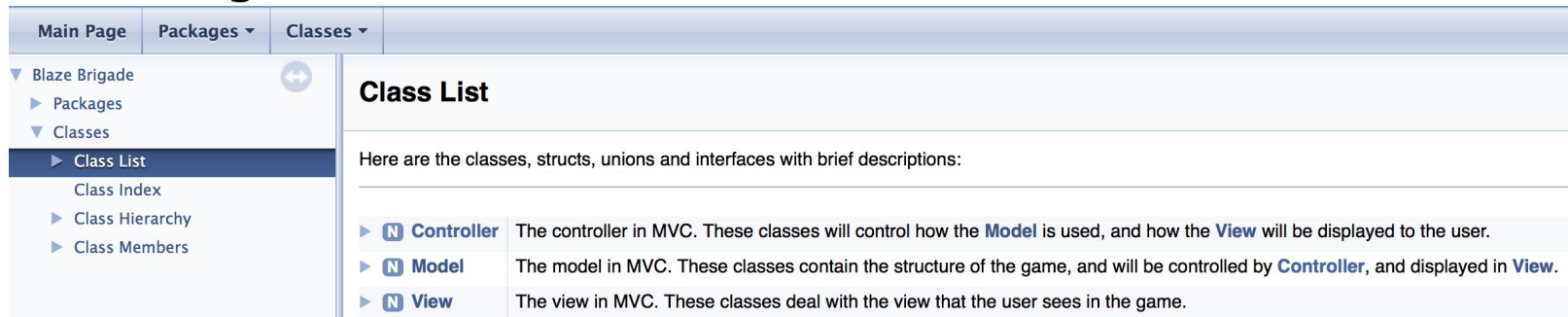
Communication

- Slack
- Skype

Design

- The system follows the MVC architecture
- Composing the modules as a Model, View and Controller
- Any addition of new classes can easily be integrated into the current system
- The Module Guide further decomposes the system into a modular structure to define design pattern and anticipated changes during development

Blaze Brigade



The screenshot shows a web application interface for 'Blaze Brigade'. At the top, there are navigation tabs: 'Main Page', 'Packages', and 'Classes'. The 'Classes' tab is active. On the left, a sidebar menu shows a tree structure: 'Blaze Brigade' (expanded) contains 'Packages' and 'Classes'. Under 'Classes', 'Class List' is selected, with sub-items 'Class Index', 'Class Hierarchy', and 'Class Members'. The main content area is titled 'Class List' and contains the text: 'Here are the classes, structs, unions and interfaces with brief descriptions:'. Below this, there is a table with three rows, each representing a class in the MVC pattern.

Class List	
Here are the classes, structs, unions and interfaces with brief descriptions:	
▶ Controller	The controller in MVC. These classes will control how the Model is used, and how the View will be displayed to the user.
▶ Model	The model in MVC. These classes contain the structure of the game, and will be controlled by Controller , and displayed in View .
▶ View	The view in MVC. These classes deal with the view that the user sees in the game.

Demo

Main Menu



Maintainability

- The system follows the MVC architecture separating model, view and controller
- This ensures that the software can be easily modified when adding new features
- Also helps the software evolve, with the addition of new classes, weapons and inventory items
- These changes would not cause the software to go under excessive changes in the source code and design

Robustness

- Ensuring the software system is stable in extreme conditions and faulty user input
- Manual testing on various machines
- Stress testing on various machines
- Inputs are encapsulated to only a mouse click, which holds the majority of the scope of testing

Feedback

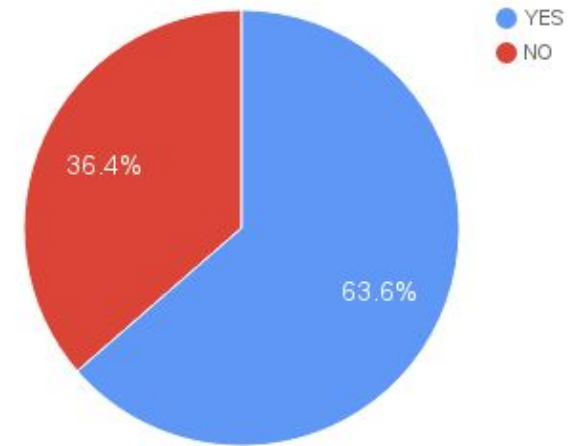
Usability Survey

73% of the responses rated 8 stars or above on the easiness of the game controls

82% of the responses rated 8 stars or above on the visuals of the game

73% of the responses rated 8 stars or above on the overall rating of the game

Would you play again?



Testimonial

“Really nice visuals, overall a great game”

“Opening screen looks great and the art in the game looks great as well. Easy to play, but not my style of game”

Next Steps

Here are the following items that would be addressed if the project continues:

- Integrating story component
- Adding missions to make the game more enjoyable
- Allowing users to go through a tutorial
- Implementing an AI for single player games
- Bootstrapping graphics and sound
- Increase level of inventory - portions, items and weapons
- New maps

Conclusion

- Blaze Brigade was implemented to provide form of good entertainment
- A tribute to the popular genre, by refurbishing to enhance user experience by integrating new features and graphical enhancements
- Gameplay
 - Moving units
 - Attacking units
 - Inventory
- Graphics
 - Main Menu
 - Animation
 - Sound

Questions?

Comments?
