**Prepared by: Team B5**

Lin, YuCHEN | NELSON, JORDAN | PARK, RYAN | WANG, XINYENG | vAN HEERDE, WILLIE

<Game Title>

Requirements Document

Version 1.1

September 30, 2016

# **Introduction**

## Team Members

Our team consists of two computer engineering students, Xingeng Wang and Yuchen Lin, as well as three computer science students: Ryan Park, a fourth year; as well as Jordan Nelson and Willie van Heerde, third years. Our objective is to create a multiplayer turn-based strategy video game.

## Purpose

<add stuff>

## Intended Audience

<add stuff>

## Project Constraints

<add stuff>

# **Description**

## 2.1 Game Summary

The game consists of a varied number of players and is played on a hexagonal tile-based board. The board’s size is dependant on the number of players playing. Each player starts on an edge tile of the board, evenly spaced from the other players. Each player also starts with three robots: a scout, a sniper, and a tank. The three robots have separate and unique attributes for health, attack damage, movement, and shooting range.

Turns are somewhat complicated to explain, but simple once the concept is grasped. Turns are separated into three categories: a turn, a round, and a play. In a turn, a player plays the robot with the highest movement points that hasn’t yet played that play. In that turn, the player can move that robot anywhere on the board (up to the maximum movement range of that robot), fire anywhere on the board once per turn (up to the maximum range of that robot), or even choose to do nothing at all and end the turn all together.

When the turn is over, the next player begins their turn, by finding their robot with the highest movement points that hasn’t yet been used that play, and so on. When all players have had one turn, the round is over. A new round begins in the same manner, with the next highest movement point robot in each player’s arsenal, and so on. When all players have moved all robots the play is over, and a new play begins. Note that if a player’s robot is killed, and they only have two left for example, they will not have a turn during the third round, as all of their robots will have been used that round.

## 2.2 Win Conditions

The objective of the game is to be the last player with robots alive, by killing or having other players kill all other robots.

## 2.3 Stalemate Conditions

As an extra feature, as well as a precaution against certain AI vs. AI games having an infinite amount of turns, we will implement a stalemate system, where after certain conditions are met (to be decided) the game will end in a stalemate.

# System Features

The nature of the game means we have many options on how to implement it, and the ability to add quite a few additional features. Features are divided into two (2) main categories: core features and additional features. Core features are essential to the game’s operation and are the bare minimum requirements we will implement upon the completion of the game. Additional features are extra functionalities. They will only be implemented as time permits.

## 3.1 Core Features

### 3.1.1 Forth Interpreter

We will implement an interpreter to communicate between various artificial intelligence programs and our game system, so that a single human player can play against several unique computer players during a single play session. In the event the user wishes to have a game run with only computer players, features will be available so the user may either view the result as quickly as possible, or enable a ‘spectator mode’ where they may watch the computer players play against each other, at their leisure.

### 3.1.2 Hot Seat Multiplayer

Human vs. Human multiplayer will be implemented using a ‘hot seat’ multiplayer method, meaning the game will be localized on a single computer where the players will take turns playing their turn at the computer.

### 3.1.3 Graphical User Interface

The game will be displayed using a very basic graphical user interface, which the player can interact with using a mouse and various buttons displayed on the screen. The game will also have a menu system outside of the game, that the player will use to choose from various options, including how many players are playing, how many human players, etc.

### 3.1.4 Display Player Statistics

We will implement a status bar on the made interfaces, allowing players to see the robots they have left alive, their health, as well as the range and movement range of their current robot.

## 3.2 Additional Features

The next few items, as explained above, are things we would like to implement, given enough time.

### 3.2.1 Animation/Enhanced Graphical User Interface

We would like to implement animations and a more advanced graphical user interface to increase the aesthetics and enhance the immersion of the game. Three-dimensional robot/game board models may also be considered.

### 3.2.2 Sound Effects

Implementation of more sounds to increase the appeal of the game and give it more polish is another feature we would like to implement.

### 3.2.3 Leaderboard/Point System

The implementation of a leaderboard and/or a point system would enhance the competitiveness of the game

### 3.2.4 Custom Game Setups

We would like to add extra features to the game to enable customized setups, which would include things such as non-regulated map sizes (2 players on a 6 player board, 6 players on a 2 player board, etc.), power-up abilities, or terrain features to the board.

### 3.2.5 Turn Timer

Another optional feature we would like to implement is a turn timer, to limit the time a player has to make a turn. The length of this would be regulated in the game set-up menu.

### 3.2.6 Tutorial

We would like to implement a tutorial of some kind, accessed through the main menu, for new players to learn the game.

### 3.2.7 Difficulty Settings

The last feature we would like to implement, given the time, is the option for a player to set a difficulty level against a computer player. This could be implemented either by giving the player a boost/deflation to their robots’ stats, or by having a pre-screened subset of AI programs, sorted by their known difficulty level.

### 3.2.8 Networked Multiplayer

We would like add additional functionality to allow network multiplayer, where each player sits at a separate computer and waits for their turn sequentially.

### 3.2.9 Hint System

The last feature we would like to implement given enough time is an optional hint system for new players, where we make an AI program read the current human player’s positions as well as all known enemy positions and gives the player hints based on what the AI would do.