Software Requirements and Design Document

For

Group “Riskier”

Version 2.0

Authors:
Bradley Hodson
Michael Styron
Antonio Vidal
Grayson Wagstaff
Wesley Watkins

1. Overview (5 points)

a. Riskier is a 2-D command & control game. The goal of the game is to control the entire map or eliminate all other players. The game is a standalone game that implements a game board with sectioned into region that the player can play against other players or AI and control resources (soldiers) to take over other regions. When players compete for regions the game enters a turned based combat system that allows the player to use their skill and resources to compete and win.

2. Functional Requirements (10 points)

Blue - Critical Feature
Black – Preferred Feature
Red - Optional Features

1. Title Screen – Must display options
    a. Start Game
        i. New Game – Starts a new game and goes to the Game Setup screen
        ii. Saved Game – Ability to load a game from a previous save file.
        iii. Network - Able to create/join games against other networked players
    b. Options - Shows relevant customization and setting options
    c. Online Leaderboard - Displays top statistics from players
        i. Google Firebase Users - Users are able to login and saved statistics
        ii. Rankings based on wins/power - shows rankings from relevant topics
    d. Exit- Ability to exit the game (gracefully)
2. Game Setup
    a. Map Selection (if multiple) - Ability to select a board for the game
    b. Character Selection - Able to select a player for the game
        i. Character Special Ability – character has a unique skill that can be used during gameplay
    c. Opponent Settings
        i. Set Difficulty - Ability to set difficulty level to Low & High
        ii. Set Play-Style - Allows players to set an AI too passive to aggressive
    d. Additional Options
        i. Turn Settings – Allows you to set different turn options (speed, benefits, and rotation)

ii. Battle Settings - Ability to set different battle options (speed, power, and starting position)

3. Game
    a. Game Board - Displays the game board chosen by player
        i. Presentation of Map – Shows the different control areas clearly
        ii. Starting Position/Moves - Clearly marks starting position for player and lets them place units
        iii. Map Control
            1. Territory Control & Power – clearly marks power on board
            2. Movement of Power - clearly shows how power moves/created/destroyed

    b. Menu Options – Ability to change relevant options
    c. Turn System
        i. Staged Turn - Turn timer shows and counts down with specified time
            1. Allocate – Allocate stage starts and completes
                a. Introduce new power - ability to add new power to the board
            2. Move - Move stage starts and completes
                a. Move power around gameboard - ability to move power around the board
            3. Battle - Battle move starts and completes
                a. Use power to battle against opponents – ability to battle opponents on contested game tiles
    d. Player Information/Inventory - Shows player status on screen
        i. Power Available – clearly shows the available power to allocate that turn.
        ii. Statistics - Shows current game statistic of player
            1. Power gain/loss - Shows power gain/loss during the game
            2. Battles won/lost - Shows battles won/loss during the game
    e. Opponent Information
        i. Power Available - Shows power available during opponent turn
        ii. Statistics
            1. Power gain/loss- Shows power gain/loss during the game

2. Battles won/lost - Shows battles won/loss during the game

f. Power -
   i. Units – unit clearly visible and able to be acted on
      1. Unit 1 - Attack Oriented – Unit has stats with attack-orientation
      2. Unit 2 – Defense Oriented – Unit has stats with defense-orientation
   ii. Player Modifiers - Modifiers earned in game apply to units

g. Game AI – AI program launches with game application
   i. Opponent – opponent makes actions relevant to turn and input

4. Battle - player is able to enter battle scene during relevant times
   a. Menu Options - Ability to change relevant options
   b. Player Board – Ability to view unit on player board
      i. Power Position
         1. Attack Position – ability to move units to attack position
         2. Support Position– ability to move units to support position
         3. Retreat Position– ability to move units to retreat position
   c. Opponent Board - Ability to view unit on opponent board
      i. Power Position
         1. Attack Position – ability to move units to attack position
         2. Support Position– ability to move units to support position
         3. Retreat Position– ability to move units to retreat position
   d. Turn System
      i. To be determined by rules
   e. Player Actions
      i. To be determined by rules
   f. Battle Information
      i. To be determined by rules

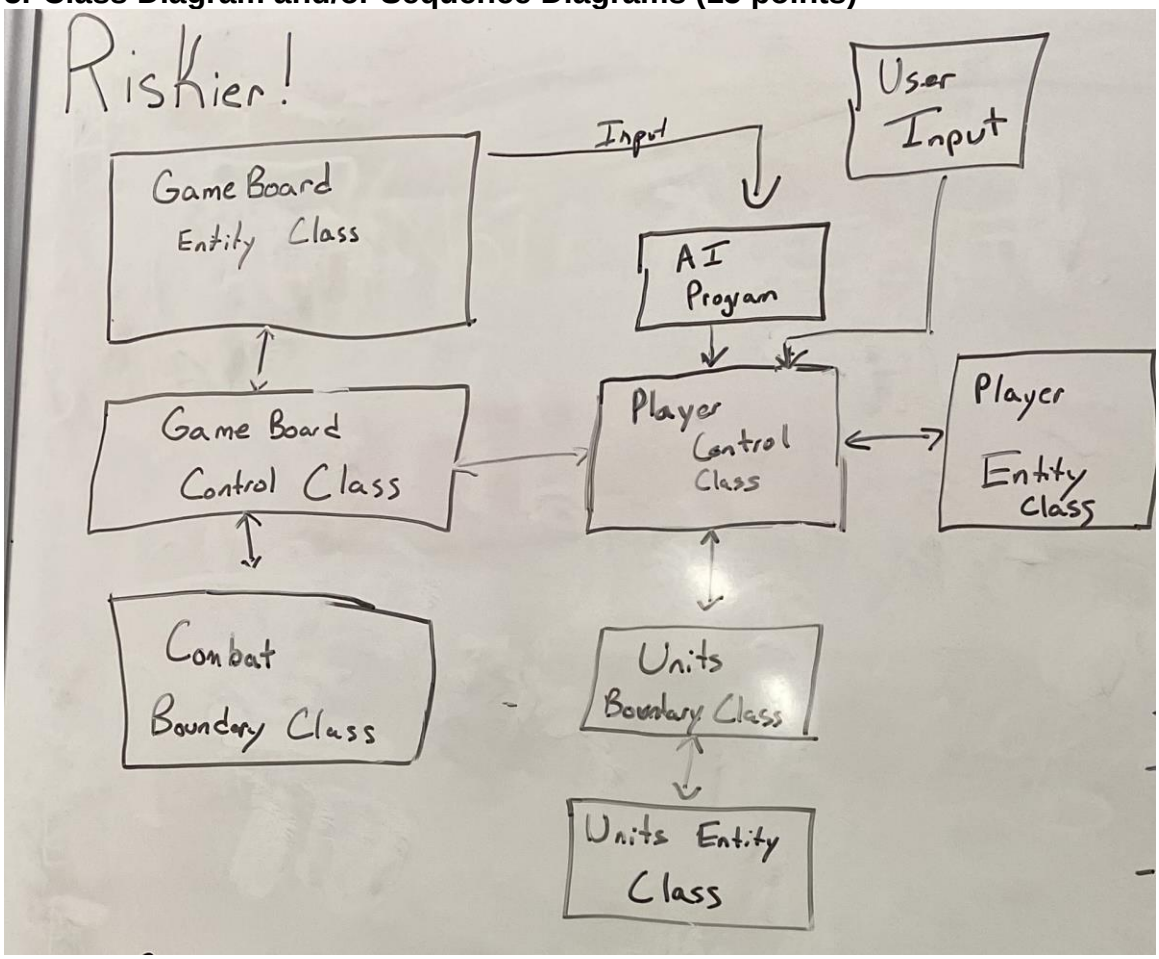
3. Non-functional Requirements (10 points).

a. The game scenario must be consistent and engaging.
    i. A non-functional requirement we have identified to create a narrative that fits the play of the game. According to research (Found Here) daily gamers stated that the scenario in a strategy game is the most important aspect to them.

4. Use Case Diagram (10 points)

a. Start New Game
b. Load Saved Game
c. Select Game Options
d. Allocate Power onto Game Board
e. Move power on Game Board
f. Contest tiles on Game Board
g. Battle opponents with power
h. Result of Battle updates Game Board
i. Game Victory

5. Class Diagram and/or Sequence Diagrams (15 points)



6. Operating Environment (5 points)

a. The operating environment is not a large issue for this project. This is due to the platforms that Unity compiles to. The other aspects of the project (Python & Google Firebase) are not platform specific either. However due to the common operating systems we will primarily test for and compile on the Windows 10 operating system.

7. Assumptions and Dependencies (5 points)

a. Unity
    i. Asset Packs for Unity
        1. to be determined
b. Python3.6
    i. TensorFlow package
    ii. Numpy package
c. Google Firebase
    i. Google Developer Account