Client Side Test Plan

User Interface:

- User Interface testing will be done manually. We can visually check for the existence of our user interface components, such as labels, text boxes, buttons, etc.
- Of particular note is to test that the user interface properly refreshes upon receiving new information.

Game Models:

- Units will be tested via simple creation tests. We create units using those very same static methods.
- Terrain will be tested in the same manner as unit creation. Terrain for any map is generated through static methods, and they are tested.
- Maps will be tested via testing our MapGenerator methods, in which we use it to create a map.

Pathfinding:

- Pathfinding will require testing on the two types of units we have in the game.
- We will utilize map generators to vary the terrain, and analyze the list of coordinates the pathfinding produces.

Communication:

- Our first test of end-to-end communication will be manually setting a one-time UDP socket at a specified address and port. We will send a manually crafted authentication packet, read the response, parse the response into the proper bytes, and display them to ensure our tests are OK.
- Once we have our packet reading framework integrated, we will test it by firing several packets of different types to see if our framework can recognize and properly parse the information.

Integration Testing:

- Arguably the biggest realm of testing for our project lies in integration testing.
- We will have a server running, and we will try to run through a networked game.
- All of our integration testing must be done manually, as we will need to set breakpoints to inspect the runtime values of variables on both the client and the server.

Server Side Test Plan

Game Models:

- Game Models will be loosely tested, as most of the invalid data will be prevented from being hit, courtesy of the client.
- Basic unit testing on model implemented to ensure static data, units, terrain, and map creation was correctly implemented.

Integration Testing:

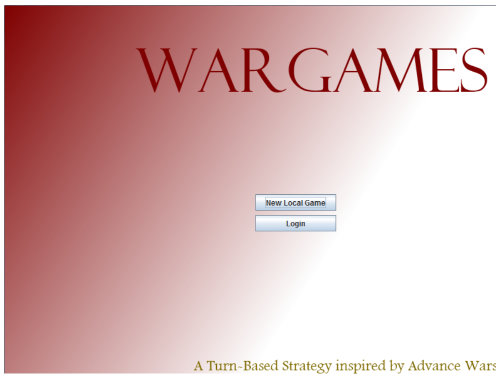
- Basic server side integration testing on the server side will simply involve creating a new game and running a pre-defined set of both valid and invalid moves, ensuring the valid moves update the model, and the invalid moves are avoided
- The rest of integration testing will involve running a client side game and manually debugging as described above

Communication:

- Like the initial communication tests of the client, the first few server tests will test the server's response to all types of packets. This will be tested with a mock client, as well as by running multiplayer test games.
- Advanced testing will include the actual client in end-to-end packet reading and writing. This will involve manually stepping through code on both client and server to ensure the two game models do not go out of sync

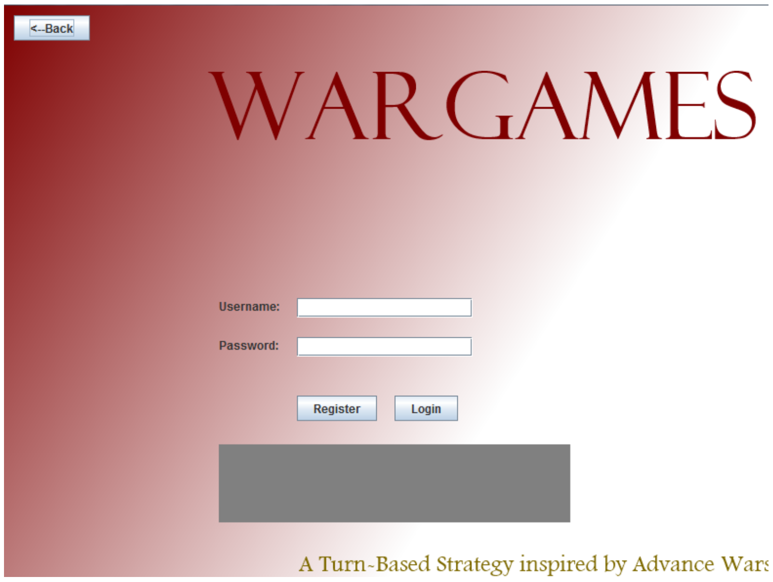
Ad-Hoc User Interface Testing Plan

Main Window



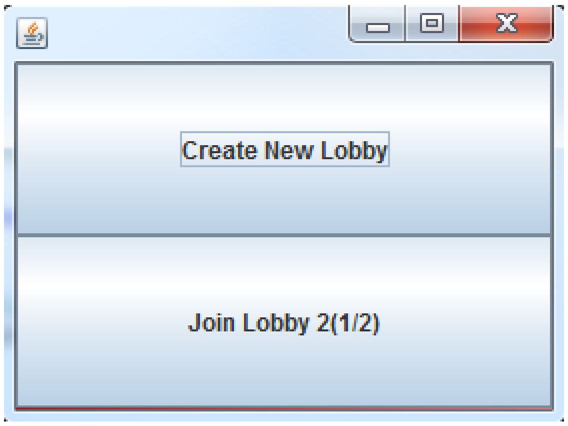
- Local Game takes user to the Game Window, setting up a Local Game.
- Login button takes user to Login Window.

Login Window



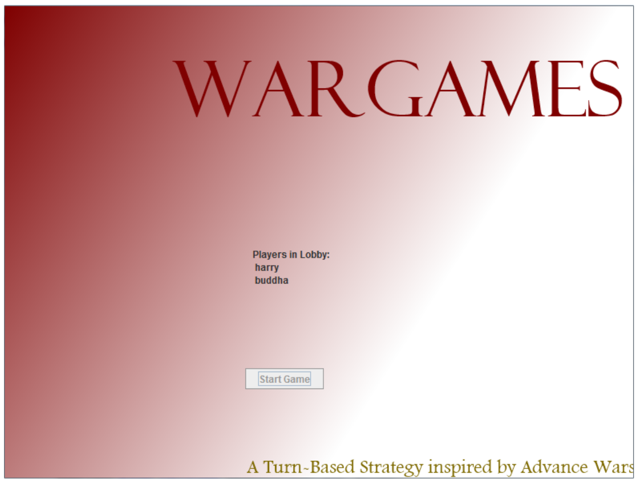
- Back Button takes user back to the Main Window
- Register registers the username and password to the server
- Login will take the user to the Lobby List Window if authentication is successful
- Responses from the server are displayed on the gray textbox.

Lobby List Window



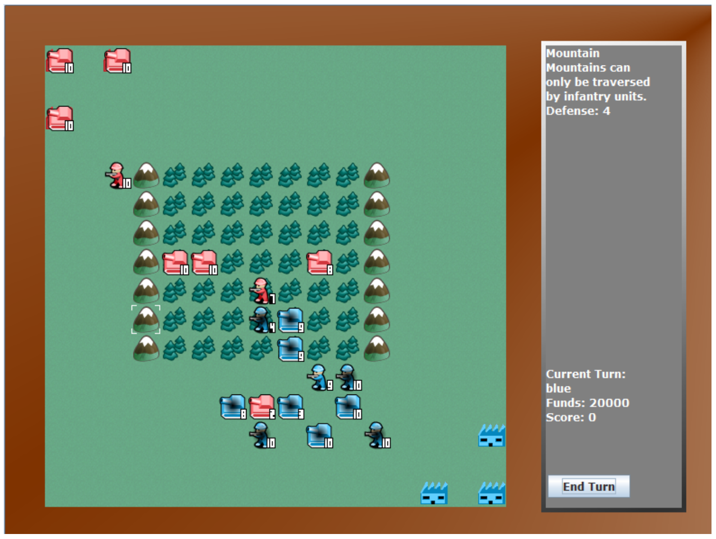
- Create new lobby will result in a new lobby created on the game server, and takes the user to the Lobby Window.
- Joining a Lobby that isn't full will take the user to the Lobby Window of that lobby.

Lobby Window



- Start Game button will remain deactivated until there are two players in the lobby, and the user is the one who created the lobby.

Game Window

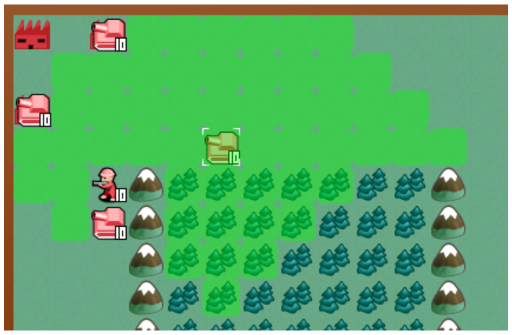


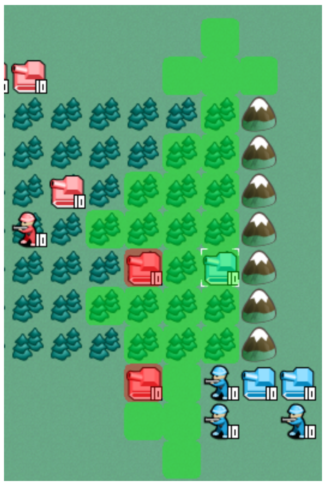
Mountain
Mountains can
only be traversed
by infantry units.
Defense: 4

Current Turn:
blue
Funds: 20000
Score: 0

- Clicking on End Turn when it is your turn will end the turn.
- Clicking on End Turn when it is not your turn will do nothing.
- Active units are shown as not-darkened
- Inactive units are shown as darkened
- Unit health is shown on the lower right corner of the unit
- Selecting on a unit that isn't inactive will highlight the terrain as follows:
  - Valid move locations are marked in Green
  - Valid enemy units that the selected unit can attack are marked in Red
  - Shown in the Pathfinding tests below
- Clicking on a Factory of your color will open a Factory Window to create units.
- Player's Funds are shown on the lower right corner.
- Current turn is shown on the lower right corner.
- When selecting on a unit or terrain (regardless of active/inactive) the information panel on the right-hand side will populate with information about the terrain and unit:
  - Terrain will have a short description, and show its defense value.
  - Unit will display a short description, its current health value, its range value, its mobility value, and if it is active or not.

Pathfinding Testing:





- Pathfinding must factor in a unit's capability to move over certain terrain.
  - Shown above, tanks cannot move across mountains.
  - Also shown above, tanks have reduced mobility walking across Wood terrain compared against simple Plains terrain.
- Pathfinding must factor in enemy units and friendly units blocking the path.