**Project Design**

**Data Structures**

Game Board: 2-Dimensional Integer Array.

The game board that contains all user activity will be represented with a 2-dimensional integer array. It is expected to be 20 by 40. Different integers will represent different game elements, and will be translated to characters when printing to the screen.

Elements:

0 – Empty Board Space

1 – Player Unit

2 – Enemy Unit

3 – Full Cover

4 – Half Cover

Translation:

0 – \_ (Underscore)

1 – X (The letter X)

2 – A (The letter A)

3 – ░ (ASCII 176)

4 – ▄ (ASCII 220)

Unit: Data Structure

Each unit controlled by the player or enemy will have a set of stats associated with them. These stats are health, location, action points, aim, grenades, and mobility.

Each team will have a separate array of units. They player team gets 6 units, the enemy team gets 10.

Name: The unit’s first and last name followed by a unit class. Unit class gives the player a hint as to how to use the unit. I.E. Grenadier, Scout, Sniper.

Health: Dictates unit’s remaining hit points, and is an integer between 5 and 10. Hit points are lost when an enemy attack is successful. When this count is zero, the unit is considered dead and can no longer be controlled by the player. When this is the case the unit will be removed from the game board.

Location: Stores unit’s current location.

Action points: All units get two action points. Moving costs 1 action, shooting costs 2 actions, throwing a grenade costs 2 actions. Action points will be processed in such a way that Units can move then shoot, but not shoot then move. If the unit has zero or fewer action points, they will be prohibited from doing any other actions until the next turn.

Aim: Aim dictates the likelihood of the unit hitting its target, and is an integer between 80 and 100. The percentage change to hit a target for all units is calculated as (Aim - 2 \* Distance – Cover Bonus). If negative, the shot will miss.

Distance is calculated as sqrt(distance in rows^2 + distance in columns^2). The largest possible value for this calculation is 14.14, as units have a sight range of 10 squares in each direction.

Cover bonus is 30 for half cover, 50 for full cover. These bonuses are granted if and only if a full or half cover tile is between the attacker and defender, and the defender is next to that tile.

Grenades: Grenades is the number of grenades the unit is carrying. Grenades destroy the environment in a 3 x 3 area where they land, and are guaranteed to do 3 damage to ALL units in range. Enemy units don’t get grenades.

Mobility: Mobility dictates how many spaces a unit can move instead of firing. Spaces are counted as the change in rows + the change in columns. Units will not move diagonally to get to a new location.

Ammo: how many shots the unit can take before reloading. Reloading costs 1 ap.

**Flow of Control**

At the start of the game a splash screen will display with the game’s logo followed by a “Press enter to continue prompt”.

\_\_\_\_\_ \_\_\_\_\_ \_\_\_\_ \_\_ \_\_

/ \_\_\_\_| / \_\_\_\_| / \_\_ \ | \/ |

| | \_\_\_\_\_\_ | | | | | | | \ / |

| | |\_\_\_\_\_\_| | | | | | | | |\/| |

| |\_\_\_\_ | |\_\_\_\_ | |\_\_| | | | | |

\\_\_\_\_\_| \\_\_\_\_\_| \\_\_\_\_/ |\_| |\_|

Press enter to Continue…

After the user starts the game, the game board populated according to a text file, board.config. The player’s squad will be filled according to a text file squad.config. The enemy squad will consist of 10 pre-defined units of identical and average strength. The player’s squad will have units with a range of attributes that change how the unit should be used.

**Player turn**

The player always gets the first turn, and must use their units sequentially as they appear in the squad array (unless skipping units and looping around until all units have no action points is feasible given time constraints). When a unit is selected, a menu will be printed below the map:

1. Move unit
2. Shoot
3. Throw Grenade

After selecting an option, relevant information will be displayed. Option one will reprint the map with available move spaces marked with a \* character, and a prompt to pick a location.

Option two will generate and display a list of targets in range and % chance to hit each in the form ENEMY (COL, ROW) %X, x being the chance to hit. After a target is chosen the unit’s chance to hit is compared against a random number generation to determine success or failure. Every unit does 4 damage ± 2 depending on a random number generation. The player will be informed as to how much damage they do.

Option three is similar to option one. The grenade throw range follows the same rules as the unit move range. After a location is picked the map is updated and redrawn to account for the grenade’s destruction.

**Enemy Turn**

The enemy will iterate through its squad sequentially. If a unit does not have any targets in range it will move toward the nearest player unit. It will try to end its turn next to the first, best cover it can while moving toward the player’s unit (this will be the most difficult part of the program. If the challenge is too great this is going to be a two-player game). Enemy units that have the option will shoot at the easiest target. After each enemy action the player will be informed by an output line and prompted to hit enter to acknowledge.

**End State**

When either side is out of units the game will end, and announce your victory or defeat.

\_\_ \_\_ \_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_\_\_ \_\_\_\_ \_\_\_\_\_ \_\_ \_\_

\ \ / / |\_ \_| / \_\_\_\_| |\_\_ \_\_| / \_\_ \ | \_\_ \ \ \ / /

\ \ / / | | | | | | | | | | | |\_\_) | \ \\_/ /

\ \/ / | | | | | | | | | | | \_ / \ /

\ / \_| |\_ | |\_\_\_\_ | | | |\_\_| | | | \ \ | |

\/ |\_\_\_\_\_| \\_\_\_\_\_| |\_| \\_\_\_\_/ |\_| \\_\ |\_|

\_\_\_\_\_ \_\_\_\_\_\_ \_\_\_\_\_\_ \_\_\_\_\_\_ \_\_\_\_\_\_\_

| \_\_ \ | \_\_\_\_| | \_\_\_\_| | \_\_\_\_| /\ |\_\_ \_\_|

| | | | | |\_\_ | |\_\_ | |\_\_ / \ | |

| | | | | \_\_| | \_\_| | \_\_| / /\ \ | |

| |\_\_| | | |\_\_\_\_ | | | |\_\_\_\_ / \_\_\_\_ \ | |

|\_\_\_\_\_/ |\_\_\_\_\_\_| |\_| |\_\_\_\_\_\_| /\_/ \\_\ |\_|