**CSCI 2312 Final Project Report**

**Chongtian Zhang**

Functions:

bool check(string s, int i)

Used to check if input position is valid or not

void readingfiles(vector<Ship>&ship)

Read in file to parameter vector

void AIships(vector<Ship>&ship)

Generate designated ships into parameter vector

bool operator==(const grid& gr1,const grid& gr2)

operator overloading in grid class, to see if two grid objects equal

ostream & operator<<(ostream& os, const grid& gr )

operator overloading in grid class, to output grid object to screen

void compare(vector<Ship>&ship,grid input,int &count)

function to check if input meets one position stored in vector

grid generate(grid &temp)

To generate a grid class object

void display(vector<Ship>&ship)

To print all information of ships in vector

Design:

This game uses two vectors to store each player’s ship information. We first read in player’s information from file then generate AI’s ship info to vector, and game starts.

The vector stores ship data type, which is a class and has 5 inherited classes, one for each kind of ship.

Grid is another class, which stores a position, mainly consists two members: vertical position and horizontal position.

Player will take first move. Type in a valid position, program will compare position with AI’s hip position and return outcome to screen.

Besides, Player can choose to quit at any player’s turn. Just enter “Q” and game ends.

Then AI would generate a random position, to compare with player’s ships positions, return outcome to screen.

Then player will take move again, but with a new position. If enter one of previous positions, program will tell user to enter new position until user do it. By the way, AI’s position will not repeat, either.

When one side’s ship count reaches 5, then that side loses. Program will show player win or lose, game ends.

When game ends (either player quit or one side wins or draw)

Program will print all ship information from both sides.

After reading and generating process are done, information will display on screen. The game automatically starts and ask for player’s input. Program will not continue until player enters guess.

The user interface will just run as described above.

Status: Every function is tested and work normally.

Testing status: It compiles and runs well on Clion and csegrid