Jacob Peterson

Rena Hofmeister

CIS 308 Final Project Proposal

For our final project, we will design an implementation of battleship with a single player facing off against a computer. The ships will be randomly placed on both players boards, and the players will take turns firing at the opponent’s ships. The user wins if they sink all of the opponents ships before their own ships are sunk. We wanted to do battleship as our final project because we felt that doing a game of some sort would be the most fun to design and implement, and we felt that battleship could easily utilize the various elements and features of C that we have learned throughout the semester.

The board would simply be printed on the command line, with both the users board and what is known of the opponents board printed and refreshed every turn. On their turn, the user would be prompted to enter the row and column they wish to guess, and then the opponents board would be updated appropriately. On the computer’s turn, the user would be informed where the computer fired at, and their own board would be updated appropriately.

Arrays would need to be used to store each ships segments in the ship structs, and which segments of the ship have been hit by the opponent. Arrays would also be used to read in the users guess of which row and column pair they wish to fire at. Strings will be necessary to print and continuously refresh both boards each turn. Pointers will need to be used to access the location of each ship in memory each turn. We will use a struct for each type of ship to be represented on the board (the ship sizes vary, so we will need 5 different structs). Also, we can use an enumeration to keep track of the turn. We will need to use dynamic memory allocation when we are changing segments of ships to being hit after the opponent hits them, as well as for when they are sunk. We may need preprocessor macros to handle smaller functions and variables, such as recognizing a hit or a miss, or guessing a random board space when it is the computer’s turn. Finally we may wish to use pointers to pointers if we need a pseudo out parameter such as when we are modifying a ship and then we want to print out the board using that modified ship.

We will need a few libraries to complete this project. The time.h library will give us the capability to generate random numbers for use in the computer’s turns. The math.h library will give us various useful mathematical functions, and the stdlib.h library will give us other general useful functions. The stdio.h library will be necessary for user input and output. Finally, the string.h library will give us string functionality.