|  |
| --- |
| Thomas Gustafson |
| Project 1 |
| Battleships |

|  |
| --- |
| Thomas Gustafson  CSC 17 44082  4-14-2019 |

**Introduction**

Title: Battleships

The game has been modified slightly but each player has 5 ships of length 5, 4, 3, 3, and 2. When each player has placed their ships on a 10 by 10 grid the game begins. The goal is to sink each of the enemies’ other ships. During each players’ turn they call out a location and if it is a hit their turn continues if it is a miss the turn ends. The game has been modified to allow up to 6 players.

**Summary**

Project size: About 300 lines in the main program and about 40 in the file setup program

Number of Variable:8 variables and 2 structures.

Number of Functions: 3

I used many things we have learned such as dynamic memory, structures, enumerated data, changing char cases, and files. One of my biggest obstacles required a decent bit of research into how to delete the output of each user so their ship location could not simply be looked at. The project took about 1.5 weeks from planning to implementation.

Pseudo code

*Initialize*

*Prompt user for number of players between 2 and 6*

*While loop to check user input*

*Open file containing board setup*

*For number of players*

*Prompt user for player name*

*Set pieces to 17 and active to true*

*For number of players*

*Setup board in player structs*

*Close file*

*For number of players*

*For 5 loops*

*If Else loops to decide size of ship*

*Prompt user for starting location’s row*

*If user input is invalid prompt to reenter*

*Convert to enum data type*

*Prompt user for starting location’s column*

*If user input is invalid prompt to reenter*

*Prompt user for ending location’s row*

*If user input is invalid prompt to reenter*

*Convert to enum datat type*

*Prompt user for ending location’s column*

*If user input is invalid prompt to reenter*

*If (rows are the same)*

*Set values to true*

*Else if (columns are the same)*

*Set values to true*

*While players active are more than 1*

*For number of players*

*Pass board to display program*

*Do while the input hits*

*Prompt user for name of player they wish to hit*

*Prompt user for location they wish to hit*

*If hit mark x*

*Else if miss mark 0*

*If player pieces is 0 mark as inactive and subtract 1 active player*

*Output victory message*

