Battle Ship

Final Project
SECTION C

Austin Dvorak

SUBMISSION DATE:

5/3/19

Problem

For the final project we were given the option of choosing either battleship, chess, or a third project. I chose battleship, as that was the game I was most familiar with and I felt would be the easiest to start. The goal of this project was to encompass all we have learned this semester in a single program.

Analysis

The requirements were simple. Generate a random board, have user input their board, play the game, and continuously update the board as it goes on. The 'computer' would randomly select a position and the user would manually input a position as a guess.

Design

To design this I knew that I would need to create methods to generate each boat individually. A method would then call those methods multiple times to generate the correct amount of ships. To make things easier on myself I designed the methods to allow me to be able to auto-generate the user map as well as the computer map so that it wouldn't have to be manually created each time. For each turn the user would be prompted and then the input saved. The user input would have to be converted to allow it to be interpreted by the board array. A simple conditional would detect hit or miss. Update_screen() is called after every turn and checks all cells on the boards to see if its a hit, miss or ship. Only the user board shows the boats, whereas the computer board hits, misses and empty slots. A variable for each player keeps track of the hits and ends the program on win or lose.

Testing

The main bulk of testing was in the ship generation. To get the ships to stay in bounds, not overlap, and give the ability to produce multiple of the same ships, a loop had to be created for each number guess to get it right. Each boundary also had to be checked to make sure that the ships stayed inside the board.

Comments

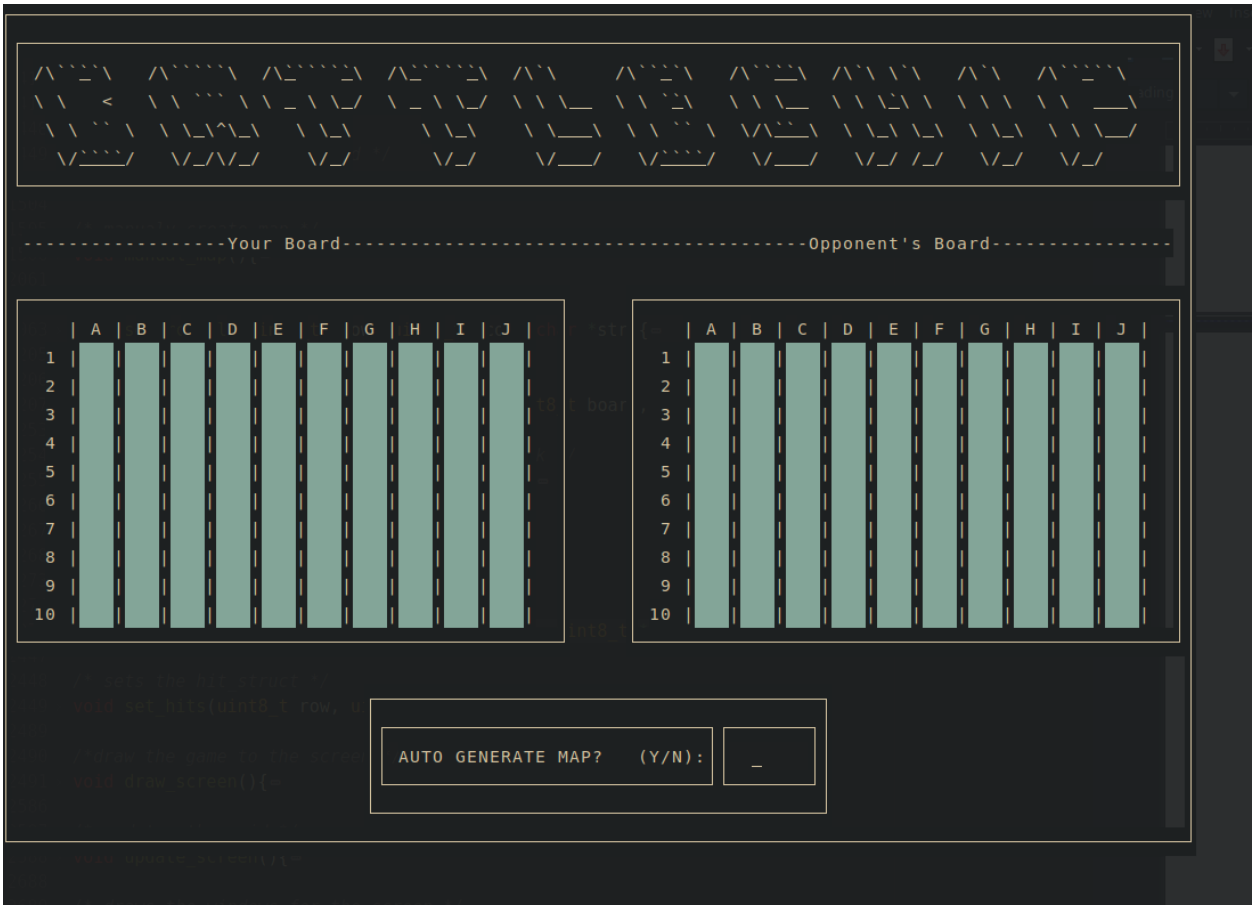
I went way too hard when creating this program. It came out to be over 3000 lines long and the computer can actually somewhat play the game. Overall this was a very fun project and really tested all of my C skills. The main methods to look in for functionality are main, update_screen(), generate_patrol(), generate_destroyer(), generate_carrier(), init_comp_board(), manual_map(), win_check(),

computer__turn(), and hit_or_miss(). Do not go into draw_title for your own safety as it serves no functional purpose and may give you an aneurysm.

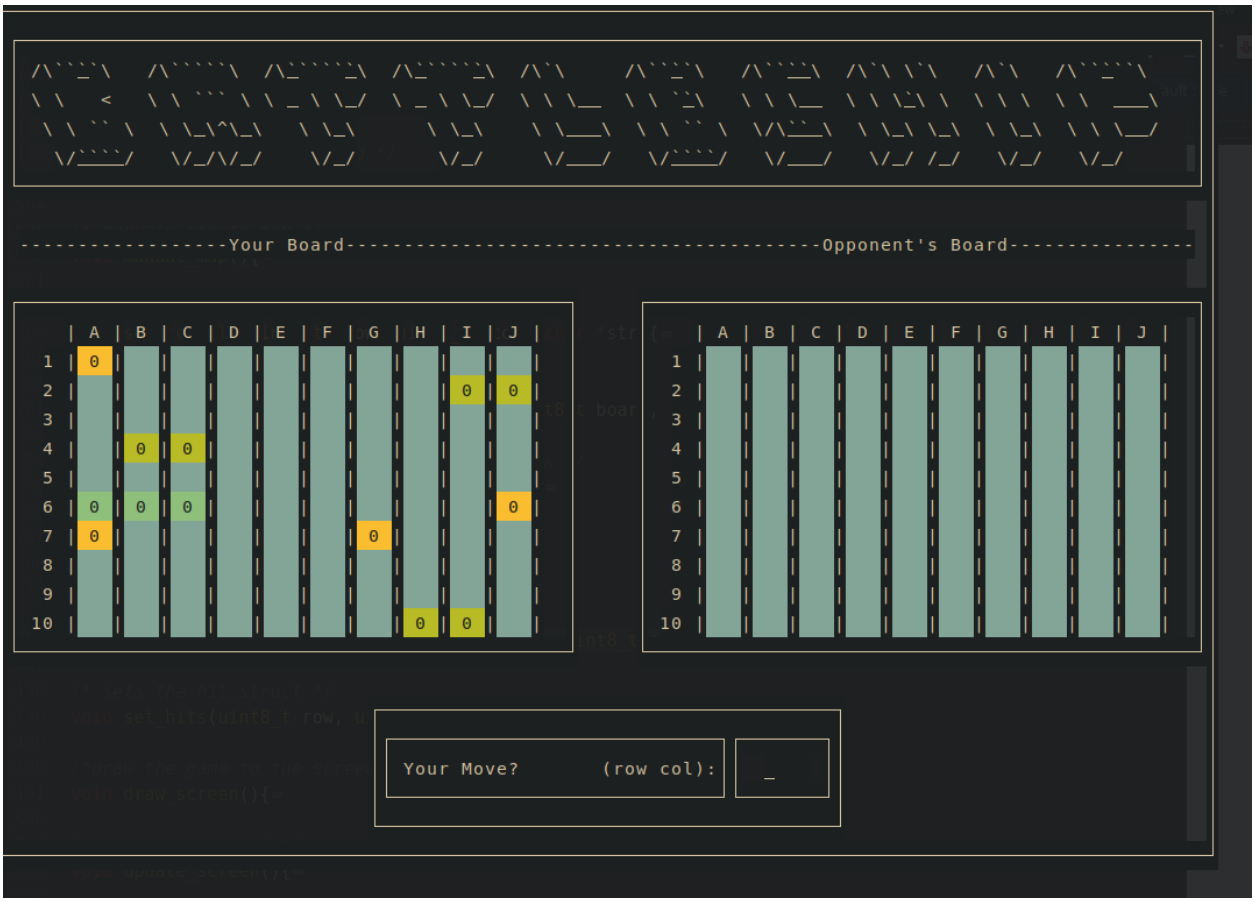
Source Code

this program is too long to screenshot, please see attached source file

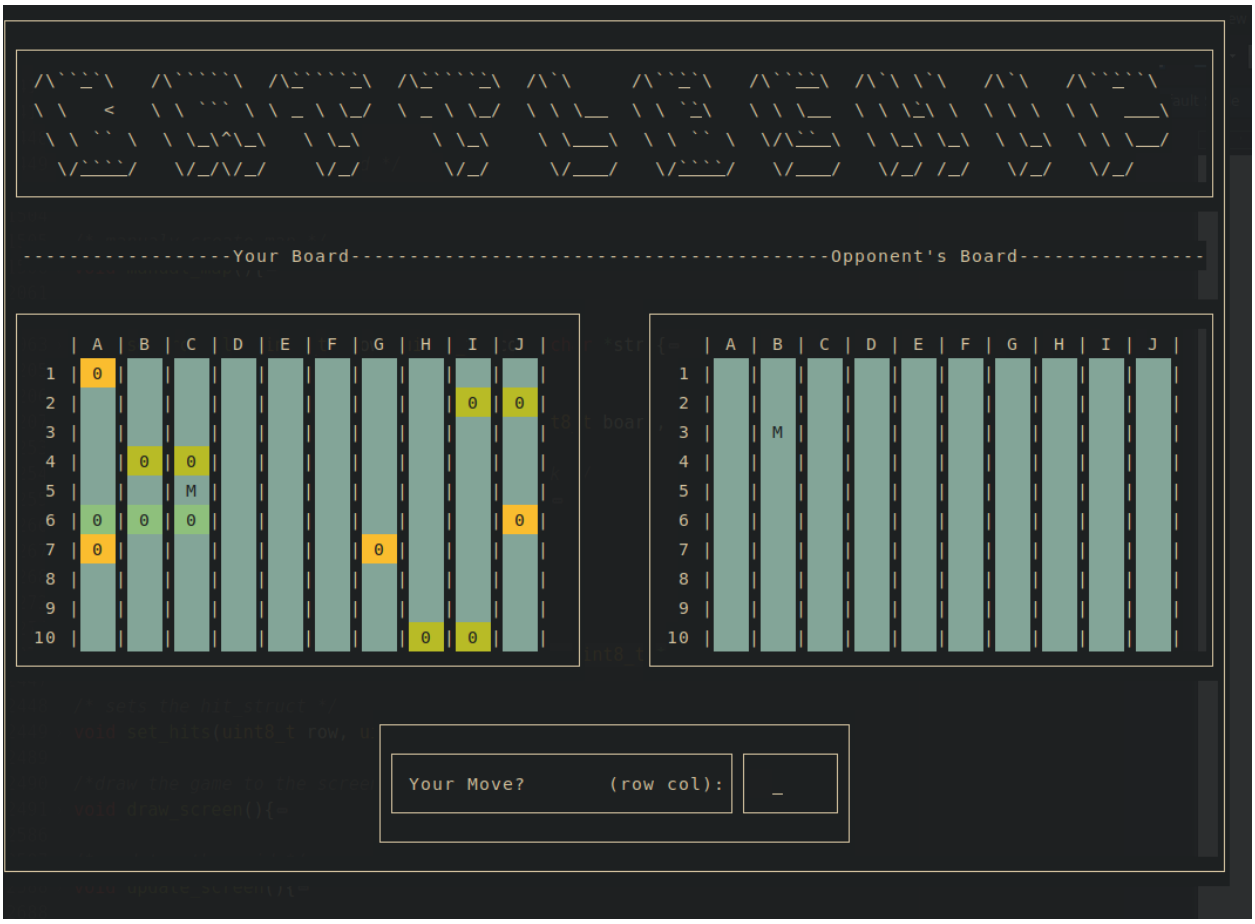
Screen Shots




Your Board
Opponent's Board
| A | B | C | D | E | F | G | H | I | J |
1 |
2 |
3 | 0 |
4 | 0 |
5 |
6 |
7 | 0 |
8 | 0 |
9 |
10 | 0 | 0 |
Destroyer 2 start (rowcol):
_


Your Board
Opponent's Board
A B C D E F G H I J
1 2 3 4 5 6 7 8 9 10
Your Move? (row col):


Your Board
Opponent's Board
| A | B | C | D | E | F | G | H | I | J |
1 | 0 | | | | | | | | | |
2 | | | | | | | | | 0 | 0 |
3 | | | | | | | | | | |
4 | | 0 | 0 | | | | | | | |
5 | | | | | | | | | | |
6 | 0 | 0 | 0 | | | | | | | 0 |
7 | 0 | | | | | | 0 | | | |
8 | | | | | | | | | | |
9 | | | | | | | | | | |
10 | | | | | | | | 0 | 0 | |
| A | B | C | D | E | F | G | H | I | J |
1 | | | | | | | | | | |
2 | | | | | | | | | | |
3 | | M | | | | | | | | |
4 | | | | | | | | | | |
5 | | | | | | | | | | |
6 | | | | | | | | | | |
7 | | | | | | | | | | |
8 | | | | | | | | | | |
9 | | | | | | | | | | |
10 | | | | | | | | | | |
Opponent's Move (row col):
C5_


Your Board
Opponent's Board
| A | B | C | D | E | F | G | H | I | J |
1 | 0 | | | | | | | | | |
2 | | | | | | | | | 0 | 0 |
3 | | | | | | | | | | |
4 | | 0 | 0 | | | | | | | |
5 | | | M | | | | | | | |
6 | 0 | 0 | 0 | | | | | | | 0 |
7 | 0 | | | | | | 0 | | | |
8 | | | | | | | | | | |
9 | | | | | | | | | | |
10 | | | | | | | | 0 | 0 | |
| A | B | C | D | E | F | G | H | I | J |
1 | | | | | | | | | | |
2 | | | | | | | | | | |
3 | | M | | | | | | | | |
4 | | | | | | | | | | |
5 | | | | | | | | | | |
6 | | | | | | | | | | |
7 | | | | | | | | | | |
8 | | | | | | | | | | |
9 | | | | | | | | | | |
10 | | | | | | | | | | |
Your Move? (row col):
_