VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

COMPUTER ARCHITECTURE (CO2007)

Report Assignment

Battleship

Advisor: Phạm Quốc Cường

Students: Cao Ngọc Lâm - 2252419

HO CHI MINH CITY, DECEMBER 2023

Contents

1 Introduction

In this report, i will demonstrate the algorithm and guide for the game Battleship

This game contains two players, each player will sets up their own fleet of battleships on a 7x7 grid. The fleet of each player must contain 3 2x1 ships, 2 3x1 ships and 1 4x1 ships. Each player will set up a ship by input 4 numbers represented the coordinates of the bow and the sterm of the ship. For example, when the player input 4 numbers "0 0 0 3", it means that the x-coordinate and y-coordinate of the bow is (0, 0) and the x-coordinate and y-coordinate of the stern is (0, 3).

After the setup section, both players will take turn to choose a cell in the opponent grid by choosing its target and attack it. If the attack hits a component of a ship in the opponent's fleet, the announcement "HIT!" would be displayed on the terminal, otherwise it will announce "MISS!". A player will lose if all of their ships got destroyed first.

2 The idea in my implementation

The main program will be divided into 4 procedures, corresponding to 4 phases of the games:

2.1 Welcome and init

This is the beginning of the game, where the program will print out welcome statement on the terminal and the essential guides for the player to understand the rules. At the same time, the program will initialize a 7x7 grid for each player in order to set their fleet. In this grid, each cell may contains value 1 or 0. The value 1 indicates that this cell is occupied by a ship, otherwise it's 0. Since there hasn't been any ship setup, so all cells of each grid contain 0:

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

2.2 Player input

In this procedure, each player will set up their own fleet. Player A will setup first. Each player will be provided 6 turns to set up their ships, corresponding to 6 ships in their fleet. In each turn, the player will input 4 numbers indicated the coordinates of the bow and the stern of the ship. For example, if the player first input the tuple "3 2 3 5", the grid would be like this:

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	1	1	1	1	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

While input, the player may encounter 4 types of errors:

1. ***Out of range error***: The number type from the keyboard is out of range from 0 to 6. For example, -1 is invalid and 10 is out of the range of the grid. In this case, the player will be alerted the out of range error and asked to input again that number.

2. ***Type error***: The 4 numbers can't form the right type of ship. For example: the input "1 2 3 4" can't form a ship, or "0 0 0 4" form a 5x1 ship, which is not the right type. This can be solved by checking if there exists an equal couple x-coordinate or y-coordinate of bow and stern. If there exists a couple like that, we need to calculate the size of the ship by getting the absolute subtracting of the two remain numbers in the tuple. For example, the size of "0 0 0 3" will be calculated by getting the absolute subtraction of 0 and 3 – it is 4. If the size is in range [2,4], the ship is in right type, otherwise ask the user to enter again.

3. ***Over amount error***: The amount of the type of the ship the player has entered is greater than the required amount. For example, the fleet has already contain 1 4x1 ship and the user enter "0 0 0 3", which will greater than the required amount 1. The user will be required to enter again the ship

4. ***Overlap error***: The ship that the user located is overlapping with the previous ship in his fleet. For example, the ship "0 2 2 2" is located before, and when the user inputs "1 2 1 4", it will cause an overlap:

0	0	1	0	0	0	0
0	0	1	1	1	0	0
0	0	1	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

This error can be handled by iterating through all the cells beginning from the new bow to the new stern. If there exists a cell containing the value 1, an error statement will be announced, and the user will be asked to enter again.

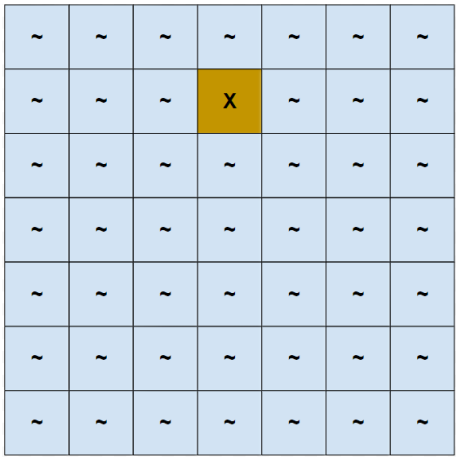
The program will handle 4 types of error sequentially in the order: out of range error, type error, over amount error and overlap error. If no error occur, an statement will be print out the terminal to indicate that the new ship located is acceptable.

2.3 Play

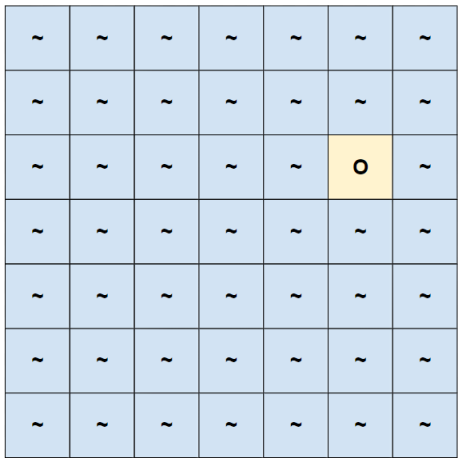
After each user successfully setup their fleet, the main game will occurred. Since there is 1 4x1 ship, 2 3x1 ships, 3 2x1 ships, there will be 16 1s in each grid. There will be a value stored in t9 to indicate the turn of the players. The value 0 in t9 indicated the turn of player A and otherwise, the value 1 indicated the turn of player B. First, the value in t9 will be 0. Implement the game loop to take turns between the players based on the value in t9. For example, after input phase, the grid of player B will be like this:

1	1	0	1	0	0	1
0	0	0	1	0	0	1
0	1	0	0	0	0	0
0	1	0	1	1	1	0
0	1	0	0	0	0	0
0	1	0	1	1	1	0
0	0	0	0	0	0	0

If player A take turns, he would input two numbers as coordinates of his target cell to fire. If that ones is occupied by a ship (contains value 1), the cells will be assigned again to have value 0 and the statement "HIT!" will be printed out to the terminal and a matrix will be shown on the screen so that player A will figure out the location of the hitting cell.



At the same time, the number of 1s in the opponent grid will be decreased by 1. If the number of 1s in the opponent grid is 0, player A will win and the game is over. Otherwise the statement "MISS!" will be printed out and change to turn of player B.



Noted that this showing grid will help user to figure out where they hit and miss the opponent's fleet, so as they can make their strategies to win the game easily. The same process will be applied for player B. If one of the player hits all the ship in the opponent's fleet, he will win the the game is over.

2.4 Exit

Print out the statement to announce the winner of the game and the game is over.

3 Flowchart

This is the flowchart illustrated my implementation above