Mohammed Khoshkhoiyazdi

July 16, 2021

CIS 200 Project 1 part 2

# Problem Statement

This is a program for super battleship game. It asks the players to place their ships on the board. They can choose the ship they want to place. Then each round they have 5 shots to shoot plus a nuclear shot. This process repeats till one of the players win.

# ReadMe *(Optional)*

# UML Class Diagram

|  |
| --- |
| Ships |
| -name: string  -hullPoint: int  -placed: bool  -shot: bool |
| +Ships()  +~ Ships()  +getName(): const string  +getHullPoint(): const int  +getPlaced(): const bool +setName(n:string): void  +setHullPoint(h:int): void  +setPlaced(p:bool): void  +setShot(s:bool): void  +getShot(): bool  +hit(): void  +*fireWeapons() : virtual void* |

|  |
| --- |
| BattleShip |
| +BattleShip ()  +~ BattleShip ()  +fireWeapons(): void |

|  |
| --- |
| Carrier |
| +Carrier()  +~Carrier()  +fireWeapons(): void |

|  |
| --- |
| Submarine |
| +Submarine()  +~Submarine()  +fireWeapons(): void |

|  |
| --- |
| Destroyer |
| +Destroyer()  +~Destroyer()  +fireWeapons(): void |

|  |
| --- |
| PTBoat |
| +PTBoat()  +~PTBoat()  +fireWeapons(): void |

|  |
| --- |
| Player |
| -name: string  -score: int  -ships: int  -tacNuke: boo  -enemy[10][10]: char  -fleet[10][10]: char  -SB: Submarine  -PT: PTBoat  -DD: Destroyer  -CV: Carrier  -BB: BattleShip |
| +Player(n:string)  +~Player()  +setShips(): void  +getFleet():ptr\_to\_arr  +getEnemy():ptr\_to\_arr  +getFleet(r:int,c:int):const char  +printBoard(board[10][10]: char):void  +checkEmpty(size: int,x: int,y: int,direction: char): bool  +validPlacement(size:int,x:int,y:int,direction:char): bool  +markShip(ship:char,x:int,y:int,direction:char): void  +converLetterToInt(letter: char): int  +validShot(x: Int,y: int): bool  +markResult(x: int,y: int, p: Player&): void  +fire(p: Player&): void  +checkIfWon(): bool  +detonate(x: int, y: int, , p: Player&): void  +checkNuke(x: int, y:int): bool |

# Test Plan

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test # | Valid / Invalid Data | Description of test | Input Value | Expected Output | Actual Output | Test Pass / Fail |
| 1 | Valid | This test shows that if the checkEmpty function works properly. This function needs to examine weather if the index that user choose is empty or not. | 1  2  H | Empty for x=1, y=2, direction=H | Empty for x=1, y=2, direction=H | Pass |
| 2 | Valid | This test examines if printBoard function works properly or not. This function needs to print out the gameboard. | Fleet board | Printing the  Flee board | Printing the  Fleet board | pass |
| 3 | Valid | This test examines if the validPlacement function works, this function checks weather the chosen spot has enough room and space based on the size of ship and direction. | 1  2  H | Correct for x=1, y=1, and direction=H | Correct for x=1, y=1, and direction=H | pass |
| 4 | valid | This test examines if the validPlacement function works, this function checks weather the chosen spot has enough room and space based on the size of ship and direction. | 8  9  H | False for x=8, y=9, direction=H | False for x=8, y=9, direction=H | pass |
| 5 | Valid | This test examines the converLetterToInt function. This function changes the char variable to int variable. | B | 1 | 1 | pass |
| 6 | invalid | This test examines the converLetterToInt function. This function changes the char variable to int variable. | U | -1 | -1 | Pass |
| 7 | Valid | This test examines the markSip function. This function marks the chosen indexes the ship mark on the fleet board. It needs to check other functions to see weather the chosen spot is empty and has enough space for the specific ship or not | X=0  Y=A  Direction=H  Ship=Carrier | Printing C in the chosen spot. | Printing C in the chosen spot. | pass |
| 8 | Valid | This test examines the markShips function. This function marks the chosen indexes the ship mark on the fleet board. It needs to check other functions to see whether the chosen spot is empty and has enough space for the specific ship or not | X=9  Y=A  Direction= V  Ship=Carrier | It does not fit! Choose other values! | It does not fit! Choose other values! | pass |
| 9 | valid | This test examines the markShips function. This function marks the chosen indexes the ship mark on the fleet board. It needs to check other functions to see whether the chosen spot is empty and has enough space for the specific ship or not | x=0  Y=A  Direction=H  Ship= Carrier  X=0  Y=A  Direction=V  Ship=Battleship | It is not empty! Choose other values | It is not empty! Choose other values | Pass |
| 10 | Valid | This test examines the placeShips function. This function ask user for row, column and direction and then checks if the chosen place is suitable for the specific ship and then print the character of the ship on the board. | X=0  Y=A  Direction=H  Ship=Carrier | Printing char ‘C’ on the board based on the row, column and direction | Printing char ‘C’ on the board based on the row, column and direction | Pass |
| 11 | Valid | This test examines the placeShips function. This function ask user for row, column and direction and then checks if the chosen place is suitable for the specific ship and then print out the character of the ship on the board. | X=0  Y=B  Direction=V | It’s not empty! Choose other values | It’s not empty! Choose other values | pass |
| 12 | valid | This test examines the placeShips function. This function ask user for row, column and direction and then checks if the chosen place is suitable for the specific ship and then print the character of the ship on the board. | X=2  Y=B  Direction=V | Printing char ‘B’ on the board based on the row, column and direction | Printing char ‘B’ on the board based on the row, column and direction | Pass |
| 13 | Valid | This test examines the validShot function. This function checks wheather the chosen position is in the array bounds or not | X=2  Y=G | Correct num | Correct num | Pass |
| 14 | Invalid | This test examines the validShot function. This function checks wheather the chosen position is in the array bounds or not | X=10  Y=A | Incorrect num | Incorrect num | pass |
| 15 | Valid | This test examines markResult function. This function checks if the chosen position hits enemy’s ship or not. If it hits, place X mark on the board and if it misses place O. | X=6  Y=D | Hit!  Place X mark on the chosen position | Hit!  Place X mark on the chosen position | pass |
| 16 | Valid | This test examines markResult function. This function checks if the chosen position hits enemy’s ship or not. If it hits, place X mark on the board and if it misses place O. | X=7  Y=H | Miss!  Place O mark on the chosen position | Miss!  Place O mark on the chosen position | Pass |
| 17 | Valid | This test examines checkIfWon function. This function checks whether the player is a winner or not. | When score I 170 | Player 1 is Victorious! | Player 1 is Victorious! | pass |
| 18 | Valid | This test examines checkIfWon function. This function checks whether the player is a winner or not. | Score less than 170 | Return False | Return  False | pass |
| 19 | Valid | This test examines fire function. This function asks the user to choose a row and column | X=0  Y=A | Hit!  Printing X in the chosen position. | Hit!  Printing X in the chosen position | Pass |
| 20 | Valid | This test examines fire function. This function asks the user to choose a row and column | X=0  Y=G | Miss!  Printing O in the chosen position. | Miss!  Printing O in the chosen position. | pass |
| 21 | Valid | This test examines fire function. This function asks the user to choose a row and column | X=0  Y=A | This index has already been taken! Please choose another spot. | This index has already been taken! Please choose another spot. | pass |
| 22 | Valid | This test examines the gameEngine. It examines all the function we have and see if everything works properly. | P1:  X=0  Y=A  Direction=H  Ship=Carrier  X=4  Y=B  Direction=V  Ship=Battleship  X=2  Y=G  Direction=H  Ship=Destroyer  X=7  Y=J  Direction=V  Ship=Submarine  X=8  Y=E  Direction=H  Ship=PTBoat | Printing C,B,D,S,and P in the chosen spots. | Printing C,B,D,S,and P in the chosen spots. | pass |
| 23 | invalid | This test examines the gameEngine. It examines all the function we have and see if everything works properly. | X=10  Y=U | Invalid! Try again  Invalid! Try again | Invalid! Try again  Invalid! Try again | pass |
| 24 | valid | This test examines the gameEngine. It examines all the function we have and see if everything works properly. | X=2  Y=B | Hit!  mark X on the chosen spot  score=10 | Hit!  mark X on the chosen spot  score=10 | pass |
| 25 | Valid | This test examines the gameEngine. It examines all the function we have and see if everything works properly. | X=0  Y=G | Miss!  Mark O on the chosen spot | Miss!  Mark O on the chosen spot | pass |
| 26 | valid | This test examines the gameEngine. It examines all the function we have and see if everything works properly. | X=0  Y=G | This index has already been taken! Please choose another spot. | This index has already been taken! Please choose another spot. | pass |
| 27 | invalid | This test examines the gameEngine. It examines all the function we have and see if everything works properly. | X=2  Y=b | Invalid Letter! Try Again: | Invalid Letter! Try Again: | pass |
| 28 | Valid | This test examines the gameEngine. It examines all the function we have and see if everything works properly. | X=8  Y=I | Hit!  Enemy PTBoat sunk! | Hit!  Enemy PTBoat sunk! | Pass |
| 29 | invalid | This test examines the gameEngine. It examines all the function we have and see if everything works properly. | X=-1  Y=A | Invalid! Try again | Invalid! Try again | pass |
| 30 | Valid | This test examines the gameEngine. It examines all the function we have and see if everything works properly. | Hit the last ship for winning | Player P1 is victorious!  Player 1 is winner! Thank you for playing | Player P1 is victorious!  Player 1 is winner! Thank you for playing | pass |
|  |  |  |  |  |  |  |

# Screenshots

Test1:



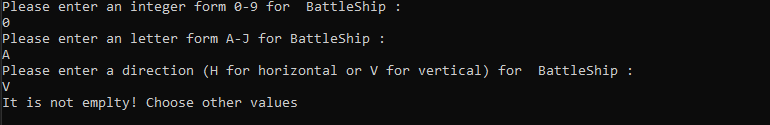
Test2:



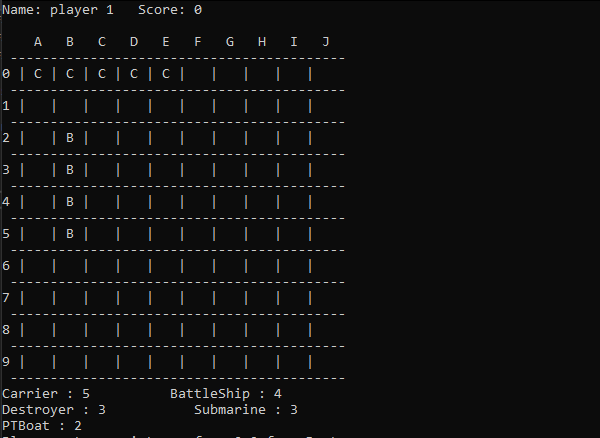
Test5:



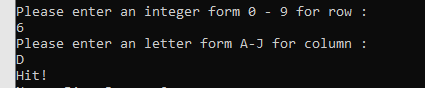
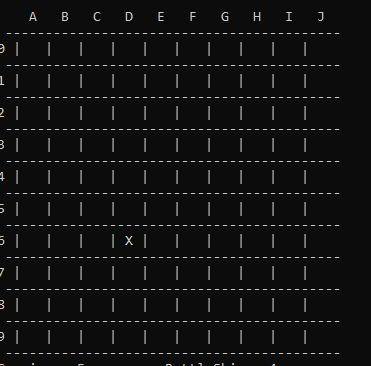
Test9:



Test12:



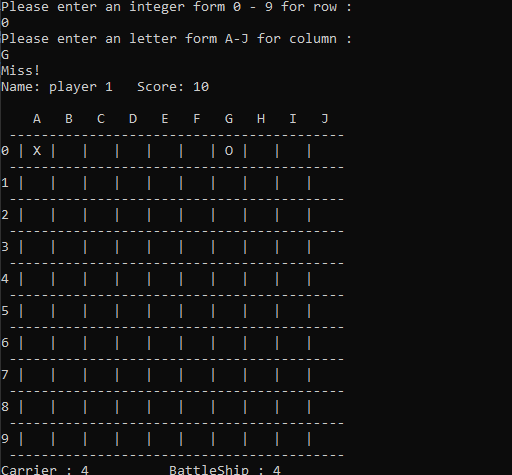
Test15:



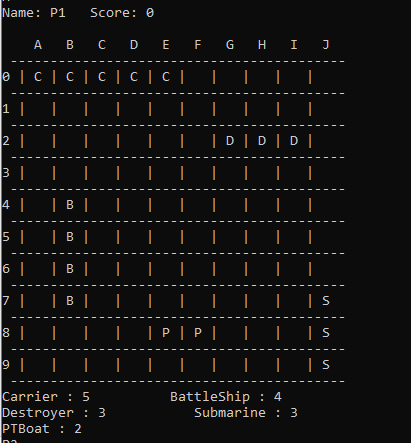
Test 17:



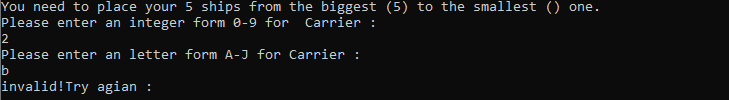
Test 20:



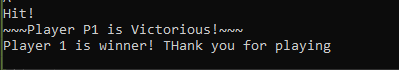
Test 22:



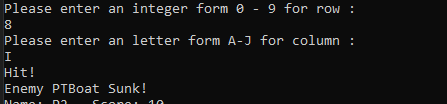
Test 27:



Test 30:



Test 28:



Part two:

# Test Plan

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test # | Valid / Invalid Data | Description of test | Input Value | Expected Output | Actual Output | Test Pass / Fail |
| 1 | valid | This test examines the fireWeapon function. This function does not take any parameters and it returns nothing. It just prints a message based on the ship typ. | CV. fireWeapon() | Launching planes for airstrike! | Launching planes for airstrike! | pass |
| 2 | valid | This test examines the setShips function. this function decrements the ships value by 1 whenever it is called. | Ships=5  Call setShips() | 4 | 4 | pass |
| 3 | invalid | This test examines the placeShips function. players need to choose which ship they want to place each time. Then function ask user for row, column and direction and then checks if the chosen place is suitable for the specific ship and then print the character of the ship on the board. | 6 | Invalid Number!  Which ship do you want to place?  1.Carrier 2. BattleShip 3. Destroyer  4. Submarine  5. PTBoat | Invalid Number!  Which ship do you want to place?  1.Carrier 2. BattleShip 3. Destroyer  4. Submarine  5. PTBoat | pass |
| 4 | Valid | This test examines the placeShips function. players need to choose which ship they want to place each time. Then function ask user for row, column and direction and then checks if the chosen place is suitable for the specific ship and then print the character of the ship on the board. | 1  0  A  V | Printing C on the chosen plots | Printing C on the chosen plots | Pass |
| 5 | valid | This test examines the placeShips function. players need to choose which ship they want to place each time. Then function ask user for row, column and direction and then checks if the chosen place is suitable for the specific ship and then print the character of the ship on the board. | 1 | You have already placed this ship.  Which ship do you want to place?  1.Carrier 2. BattleShip 3. Destroyer  4. Submarine  5. PTBoat | You have already placed this ship.  Which ship do you want to place?  1.Carrier 2. BattleShip 3. Destroyer  4. Submarine  5. PTBoat | Pass |
| 6 | valid | This test examines the checkNuke function. this function checks if the chosen plot for nuke shot is fit in the board or not. | 5  7 | Can fit in the board | Can fit in the board | pass |
| 7 | valid | This test examines the checkNuke function. this function checks if the chosen plot for nuke shot is fit in the board or not. | 1  5 | Cannot be fit in the board | Cannot be fit in the board | pass |
| 8 | valid | This test examines the fire function. it asks user if they want to use their nuke shot when they have less than 3 ships and they have Carrier or Submarine ship. Then it asks for the directions to place the nuke on the board. Then it asks users five times to shoot a shot for each ship. | Y  5  G | Printing X and O on the chosen spots | Printing X and O on the chosen spots | pass |
| 9 | valid | This test examines the fire function. it asks user if they want to use their nuke shot when they have less than 3 ships and they have Carrier or Submarine ship. Then it asks for the directions to place the nuke on the board. Then it asks users five times to shoot a shot for each ship. | N | Nuke is on standby. | Nuke is on standby. | pass |
| 10 | invalid | This test examines the fire function. it asks user if they want to use their nuke shot when they have less than 3 ships and they have Carrier or Submarine ship. Then it asks for the directions to place the nuke on the board. Then it asks users five times to shoot a shot for each ship. | Y or n | Invalid Letter! Do you want to use your Nuke Shot? (Y/ N) | Invalid Letter! Do you want to use your Nuke Shot? (Y/ N) | pass |
| 11 | valid | This test examines the fire function. it asks user if they want to use their nuke shot when they have less than 3 ships and they have Carrier or Submarine ship. Then it asks for the directions to place the nuke on the board. Then it asks users five times to shoot a shot for each ship. | Y  0  A | It does not fit in the board! Please choose other values: | It does not fit in the board! Please choose other values: | pass |
| 12 | valid | This test examines the fire function. it asks user if they want to use their nuke shot when they have less than 3 ships and they have Carrier or Submarine ship. Then it asks for the directions to place the nuke on the board. Then it asks users five times to shoot a shot for each ship. A massage should be printed before each ship shoots its shot. | Carrie did shoot its shot. Now its Battleship turn. | BattleShip  Firing 16inch guns!This is gonna be loud! | BattleShip  Firing 16inch guns!This is gonna be loud! | pass |
| 13 | Valid | This test examines the fire function. it asks user if they want to use their nuke shot when they have less than 3 ships and they have Carrier or Submarine ship. Then it asks for the directions to place the nuke on the board. Then it asks users five times to shoot a shot for each ship. A massage should be printed before each ship shoots its shot. | Battleship is sunk. Carrier did shoot his shot. Now its Destroyer turn | Destroyer  Launching depthcharges!Boom goes the dynamite! | Destroyer  Launching depthcharges!Boom goes the dynamite! | Pass |
| 14 | Valid | This test examines detonate function. this function checks if the chosen plots are empty or there is a ship over there and based on the emptiness it prints X or O on the plots. | 6  5 | Printing O on the chosen plots | Printing O on the chosen plots | pass |
| 15 | valid | This test examines the gameEngine. It examines all the function we have and see if everything works properly. | Placing ships  1  1  A  V  3  0  B  V | Printing out C and D on the board | Printing out C and D on the board | pass |
| 16 | valid | This test examines the gameEngine. It examines all the function we have and see if everything works properly. | fire  4  A | Hit!  Enemy Carrier sunk! | Hit!  Enemy Carrier sunk! | pass |
| 17 | valid | This test examines the gameEngine. It examines all the function we have and see if everything works properly. | Nuke shooting  Y  4  C | Printing O and X on the board based on the result of shot | Printing O and X on the board based on the result of shot | pass |
| 18 | valid | This test examines the gameEngine. It examines all the function we have and see if everything works properly. | Placing ships  (Already placed Battleship.)  2 | You have already placed this ship.  Which ship do you want to place?  1.Carrier 2. BattleShip 3. Destroyer  4. Submarine  5. PTBoat | You have already placed this ship.  Which ship do you want to place?  1.Carrier 2. BattleShip 3. Destroyer  4. Submarine  5. PTBoat | pass |
| 19 | Valid | This test examines the gameEngine. It examines all the function we have and see if everything works properly. | Battleship and Destroyer are sunk.  Carrier did shoot its shot. | Submarine  Bearing, set, match: Firing torpedos! | Submarine  Bearing, set, match: Firing torpedos! | pass |
| 20 |  | This test examines the gameEngine. It examines all the function we have and see if everything works properly. | 2  D | ~~~Player P1 is Victorious!~~~ | ~~~Player P1 is Victorious!~~~ | pass |

# Screenshots

Test1:

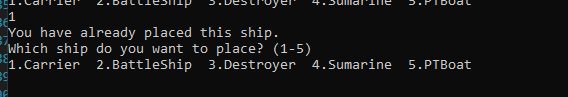


Tes2:

  
Test 4:



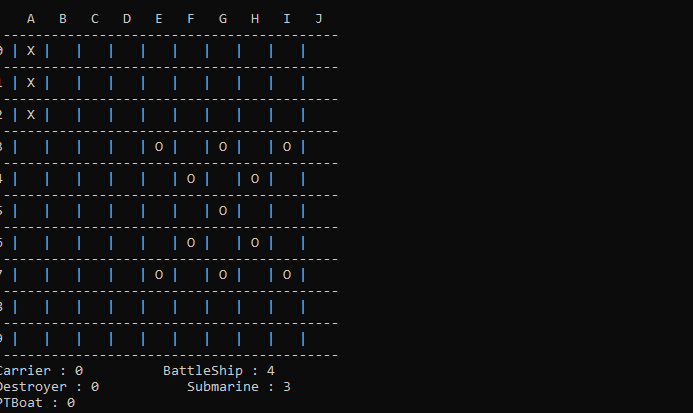
Test 5:



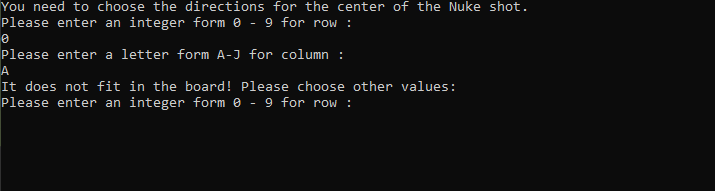
Test 7:



Test 8:



Test 11:



Test 12:



Test 13:



Test 19:



# Program Status

The program works properly and there is no error.