Farnaz Tabrizi Software Engineering Assignment 1

|  |
| --- |
| BattleShip Game |
| +CARRIER : final int  +SUBMARINE: final int  -Board: 2d Array |
| -setBoard(): void |

|  |
| --- |
| <<interface>>  SearchStrategy |
| + search(): void  + sortShipsLocation: void |

|  |
| --- |
| RandomSearch |
| -board: 2d int Array  -isSearchedCell: Boolean  -carrierLocation: int Array  -submarineLocation: int Array  -count: int  -time: double |
| +RandomSearch(int[][])  +search(): void  +sortShipsLocation(): void  -RandomNumber(int,int):int  +toString(): String |

|  |
| --- |
| SizeSearch |
| -board: 2d int Array  -isSearchedCell: Boolean  -carrierLocation: int Array  -submarineLocation: int Array  -count: int  -time: double |
| +SizeSearch(int[][])  -findCompletely(int,int,int)  +search(): void  +SortShipsLocation(): void  +toString:String |

**UML Sketch:**

|  |
| --- |
| Horizontal Search |
| -board: 2d int Array  -isSearchedCell: Boolean  -carrierLocation: int Array  -submarineLocation: int Array  -count: int  -time: double |
| +HorizontalSearch(int[][])  +search(): void  +sortShipsLocation(): void  +toString(): String |

**Journal:**

This program implemets 3 different search strategies including Horizontal Search, Random Search and Strategic Search (it is called Size Search in my program). Horizontal search starts from the point [0,0] on the game board until it finds both ships. Random search generates a random coordinates, check if we have already visited this coordinates if not it adds the coordinates to visited, and continue until it finds both ships. Size Search follows horizontal search algorithm, however it does not check the cells one by one, it checks only the third cell of every 3 cells until it finds the submarine. (it checks the first cell, then it checks the 4th cell and so on until it gets to end of the row, in the 2nd column it checks the 2nd cell, then goes to 5th cell and so on) then it checks only the 5th cell of every 5 cells until it finds the carrier.