

Draft user Manual

Battleships



Version 1
18 October 2014

Index

1. Aim of the game
2. How to execute the file
3. Instructions
4. Sample play
5. The AI

Aim of the game

Battleship (also Battleships or Sea Battle) is a guessing game for two players. It is known worldwide as a pencil and paper game which dates from World War I. It was published by various companies as a pad-and-pencil game in the 1930s, and was released as a plastic board game by Milton Bradley in 1967.

The game is played on three mazes, one for each player. The mazes are square - usually 10×10 - and the target in the maze are identified by 2 number. On one maze the player can see his ships and on the other he can record the shots by the opponent and himself.

Before each game begins, each player secretly knows the position of ships on their primary maze. Each ship occupies a number of consecutive blocks on the maze, arranged either horizontally or vertically. The ships cannot overlap (i.e., only one ship can occupy any given block in the maze).

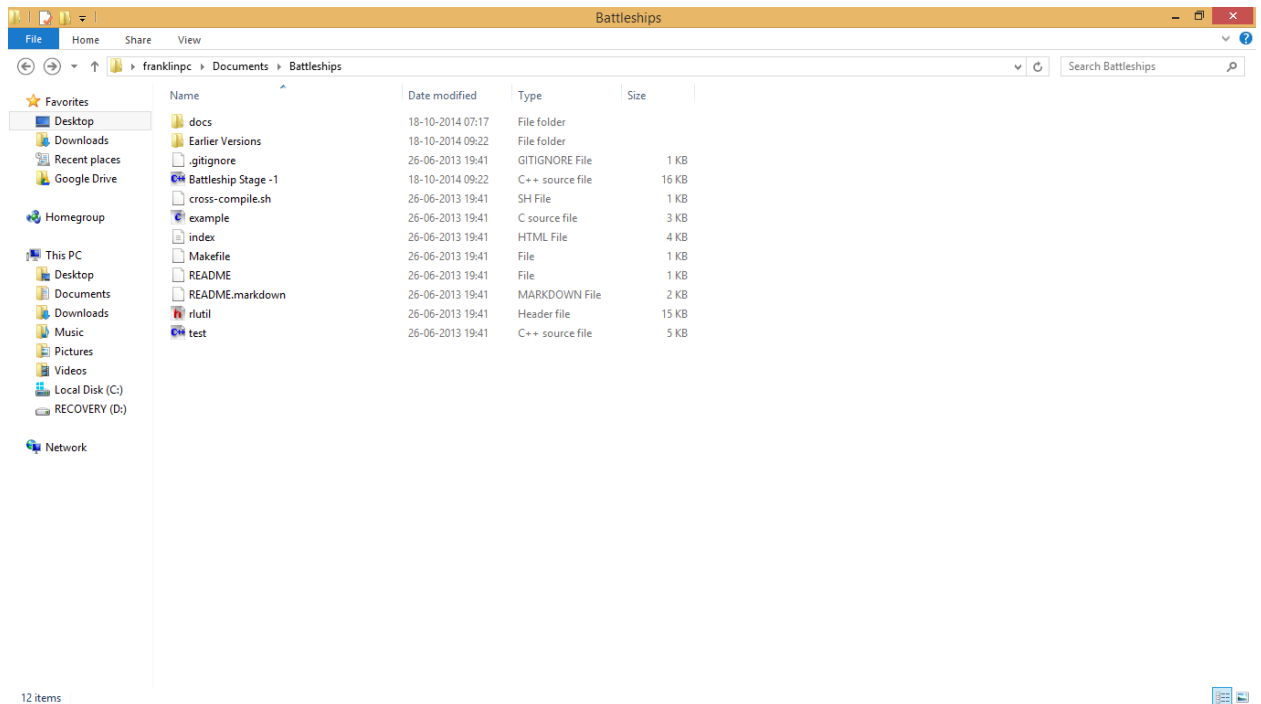
The game proceeds in a series of rounds. In each round, each player takes a turn to input a target square in the opponent's maze which is to be shot at. The Hit matrix shows whether or not the square is occupied by a ship, and if it is a "hit" the computer marks this on the maze with a specific color code. This helps in order to build up a picture of the opponent's fleet.

When all of the squares of a ship have been hit, the ship is sunk, and the Hit matrix shows the destroyed ship. If all of a player's ships have been sunk, the game is over and the opponent wins.

Source : Wikipedia (with a few modifications)

How to execute the file

1. Download the zip file.
2. navigate to Battleships Stage 1



3. Open it in code::blocks.

4. compile and execute from menubar.

The screenshot shows the Code::Blocks IDE interface. The main editor window displays the source code for 'BattleShip Stage -1.cpp'. The code includes preprocessor directives, a main function, and a series of array assignments. A console window is open, showing the output of the program, which includes a title 'Battle Ships !', a description of the game, a list of rules, and a prompt to continue playing. The 'Logs & others' window on the right shows the build log, indicating that the program was successfully compiled and executed without errors or warnings.

```
1 // A
2
3 #inc
4 #inc
5 #inc
6 #inc
7 #inc
8 #inc
9 usin
10
11 #def
12
13 //fu
14
15 //ch
16 void
17
18
19
20
21
22 a[1][1]=1;
23 a[1][2]=1;
24 a[1][3]=1;
25
26 a[8][6]=1;
27 a[8][8]=1;
28 a[8][7]=1;
29
30 a[8][1]=1;
31 a[8][2]=1;
32 a[8][3]=1;
33
34 a[2][1]=1;
```

Output:

```
Battle Ships !

This program will help you to play a well known pass-time the Battleships.
The basic aim of the game is to kill all the opponents ships before he
kills your's. The players play chance by chance, to make the game more
interesting none of them know where the other player's ship is so they have
to guess where the other players ship could be and start randomly hitting

The main things to remember are :
1. There are 4 ships for each player
2. The ships are placed on a 10x10 matrix
3. The ships are placed horizontally or vertically
4. The bombs have to be placed using co-ordinates like (0 0), (1 1)
5. Once a player hits the opponents ship or misses it a color coded
symbol is used to denote it
6. After the player's turn the computer will hit a block and it
shown in Hit maze

Do you want to continue and play (Y/N) : _
```

Build log:

```
mingw32-g++.exe -o "C:\Users\franklinpc\Documents\Battleships\BattleShip Stage -1.exe" -o "C:\Users\franklinpc\Documents\Battleships\BattleShip Stage -1.o"
mingw32-g++.exe -o "C:\Users\franklinpc\Documents\Battleships\BattleShip Stage -1.exe" "C:\Users\franklinpc\Documents\Battleships\BattleShip Stage -1.o"
Process terminated with status 0 (0 minute(s), 3 second(s))
0 error(s), 0 warning(s) (0 minute(s), 3 second(s))

Checking for existence: C:\Users\franklinpc\Documents\Battleships\BattleShip Stage -1.exe
Executing: C:\Program Files (x86)\CodeBlocks\cb_console_runner.exe "C:\Users\franklinpc\Documents\Battleships\BattleShip Stage -1.exe" (in C:\Users\franklinpc\Documents\Battleships)
```

Instructions

1. On the start of the program a few instruction have been given on how to play the program.
2. After reading the following points can be made :
 1. There are 4 ships for each player
 2. The ships are placed on a 10X10 matrix
 3. The ships are placed horizontally or vertically
 4. The bombs have to be placed using co-ordinates like eg : 0 1 which means the 0th row and 1st coloumn
 5. Once a player hits the opponents ship or misses it a color coded symbol is used to denote it
 6. After the player's turn the computer will hit a block and it shown in Hit maze
3. After you press y to continue you will see two mazes one which is the player's and one which is the hit maze which will show the places where the players played and missed and also where the player's or the computer's ship died.
4. The empty spaces are denoted by " " which says that the space are empty.
5. the computer i.e the host will ask you to enter your choice of coordinates where you want to bomb and the format is

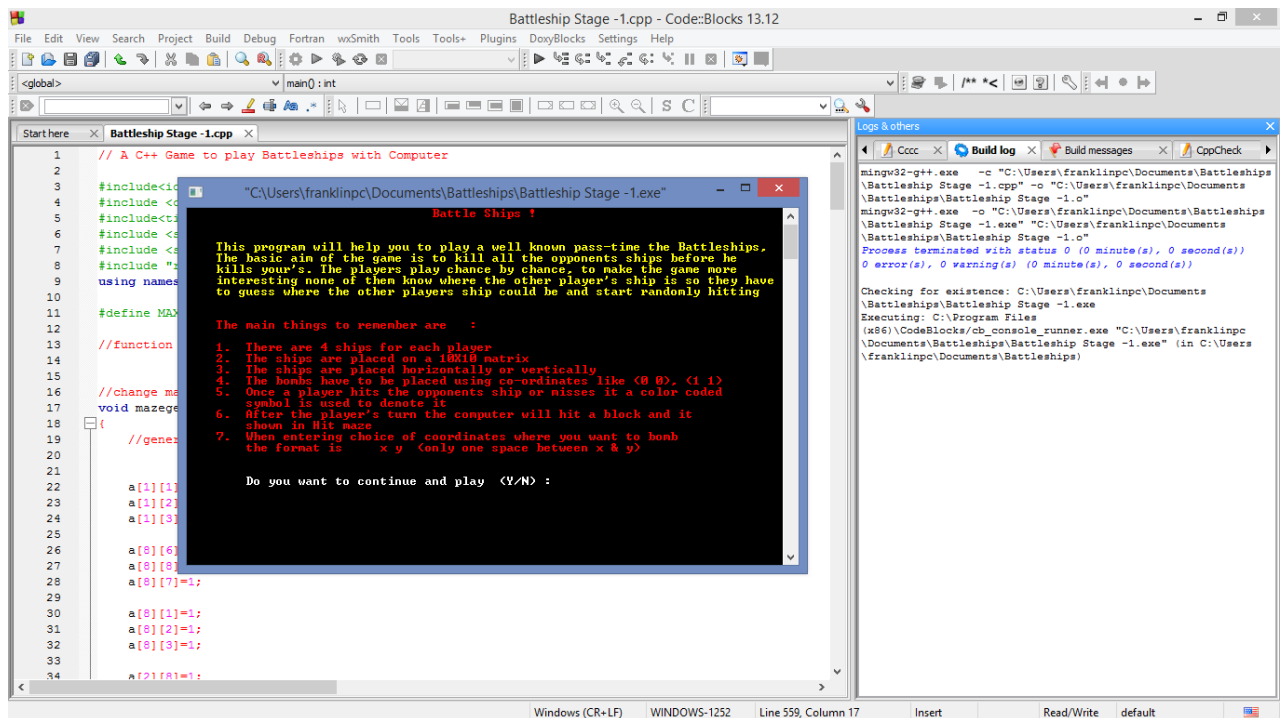
x y

note: there is one space between x and y

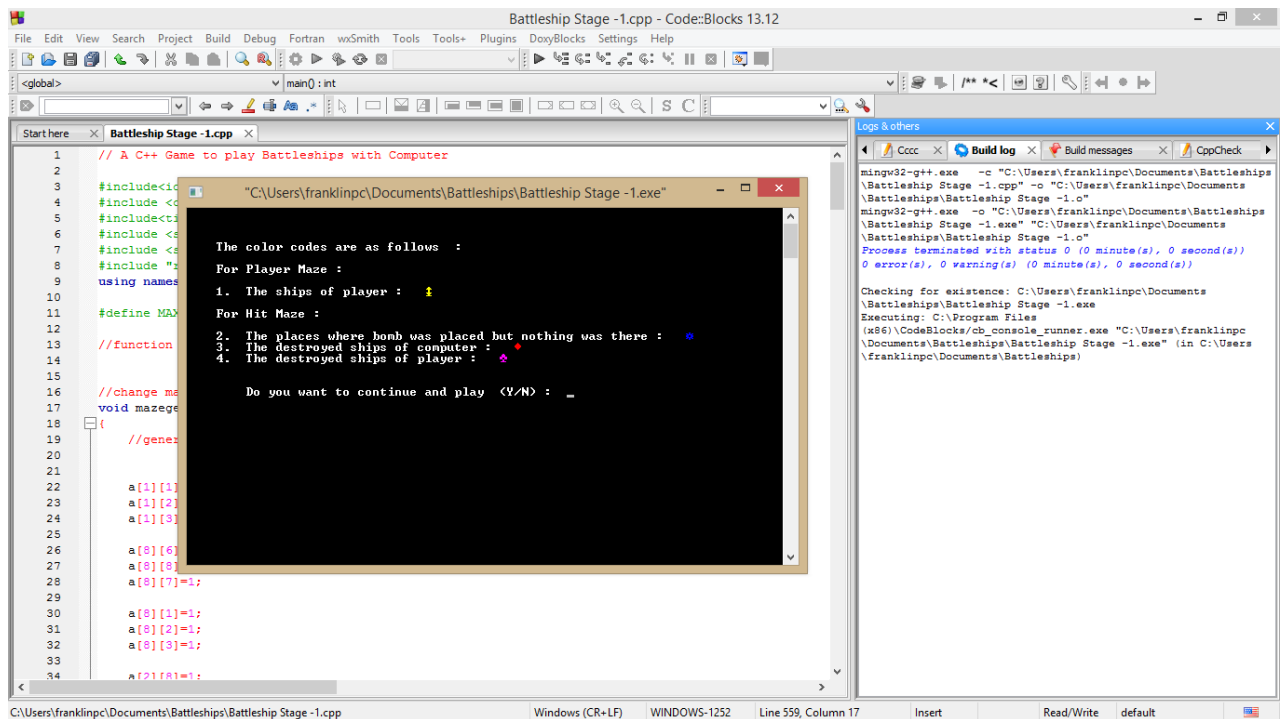
6. The game get over when If all of a player's ships have been destroyed.

Sample Play

1. Compile and run



2. Press 'y' to continue

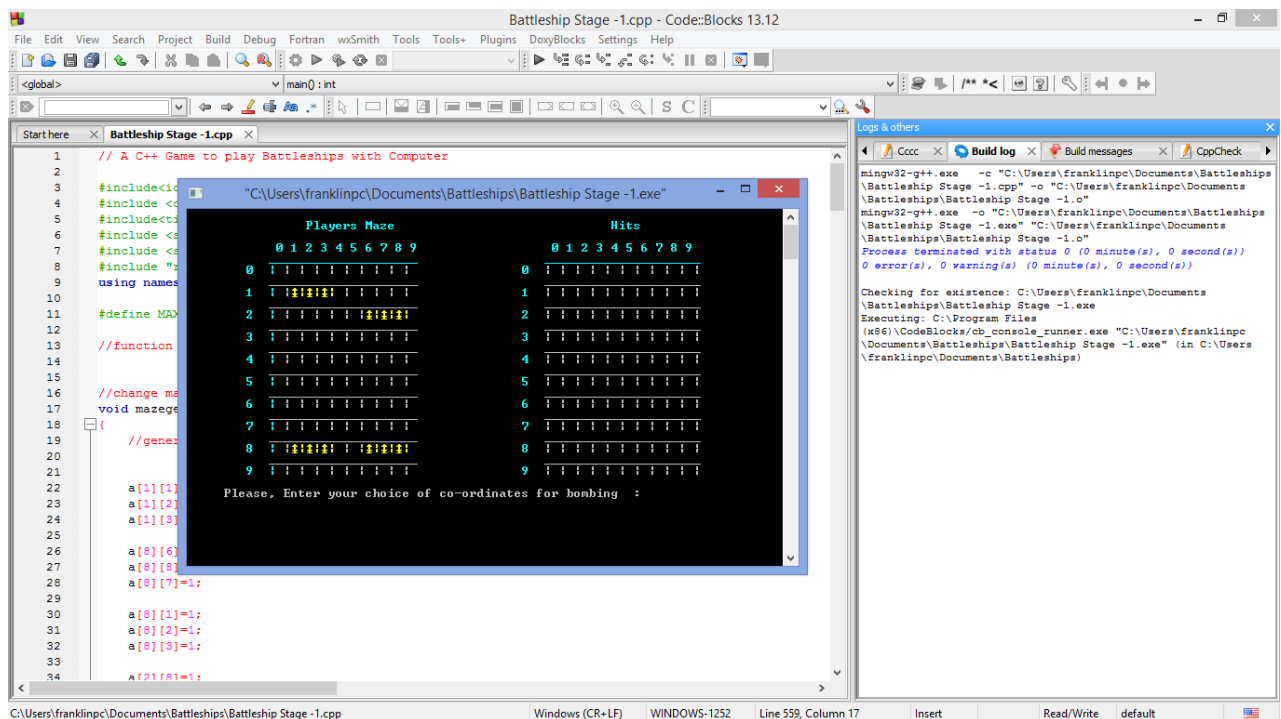


The screenshot shows the Code::Blocks IDE with the file 'Battleship Stage -1.cpp' open. The console window displays the following text:

```
The color codes are as follows :  
For Player Maze :  
1. The ships of player :  
For Hit Maze :  
2. The places where bomb was placed but nothing was there :  
3. The destroyed ships of computer :  
4. The destroyed ships of player :  
  
Do you want to continue and play <Y/N> :
```

The background code in the editor shows the initialization of the game board and the generation of the maze.

3. Press 'y' again to continue



The screenshot shows the Code::Blocks IDE with the file 'Battleship Stage -1.cpp' open. The console window displays the following text:

```
Players Maze  
0 1 2 3 4 5 6 7 8 9  
0 | | | | | | | | | |  
1 | | | | | | | | | |  
2 | | | | | | | | | |  
3 | | | | | | | | | |  
4 | | | | | | | | | |  
5 | | | | | | | | | |  
6 | | | | | | | | | |  
7 | | | | | | | | | |  
8 | | | | | | | | | |  
9 | | | | | | | | | |  
  
Hits  
0 1 2 3 4 5 6 7 8 9  
0 | | | | | | | | | |  
1 | | | | | | | | | |  
2 | | | | | | | | | |  
3 | | | | | | | | | |  
4 | | | | | | | | | |  
5 | | | | | | | | | |  
6 | | | | | | | | | |  
7 | | | | | | | | | |  
8 | | | | | | | | | |  
9 | | | | | | | | | |  
  
Please, Enter your choice of co-ordinates for bombing :
```

The background code in the editor shows the generation of the maze and the prompt for the user to enter coordinates.

4. I write a sample coordinates

```
mingw32-g++.exe -c "C:\Users\franklinp\Documents\Battleships\Battleship Stage -1.cpp" -o "C:\Users\franklinp\Documents\Battleships\Battleship Stage -1.o"
mingw32-g++.exe -o "C:\Users\franklinp\Documents\Battleships\Battleship Stage -1.exe" "C:\Users\franklinp\Documents\Battleships\Battleship Stage -1.o"
Process terminated with status 0 (0 minute(s), 0 second(s))
0 error(s), 0 warning(s) (0 minute(s), 0 second(s))

Checking for existence: C:\Users\franklinp\Documents\Battleships\Battleship Stage -1.exe
Executing: C:\Program Files\CodeBlocks\cb_console_runner.exe "C:\Users\franklinp\Documents\Battleships\Battleship Stage -1.exe" (in C:\Users\franklinp\Documents\Battleships)
```

5. here after pressing enter the computer also plays its chance

```
mingw32-g++.exe -c "C:\Users\franklinp\Documents\Battleships\Battleship Stage -1.cpp" -o "C:\Users\franklinp\Documents\Battleships\Battleship Stage -1.o"
mingw32-g++.exe -o "C:\Users\franklinp\Documents\Battleships\Battleship Stage -1.exe" "C:\Users\franklinp\Documents\Battleships\Battleship Stage -1.o"
Process terminated with status 0 (0 minute(s), 0 second(s))
0 error(s), 0 warning(s) (0 minute(s), 0 second(s))

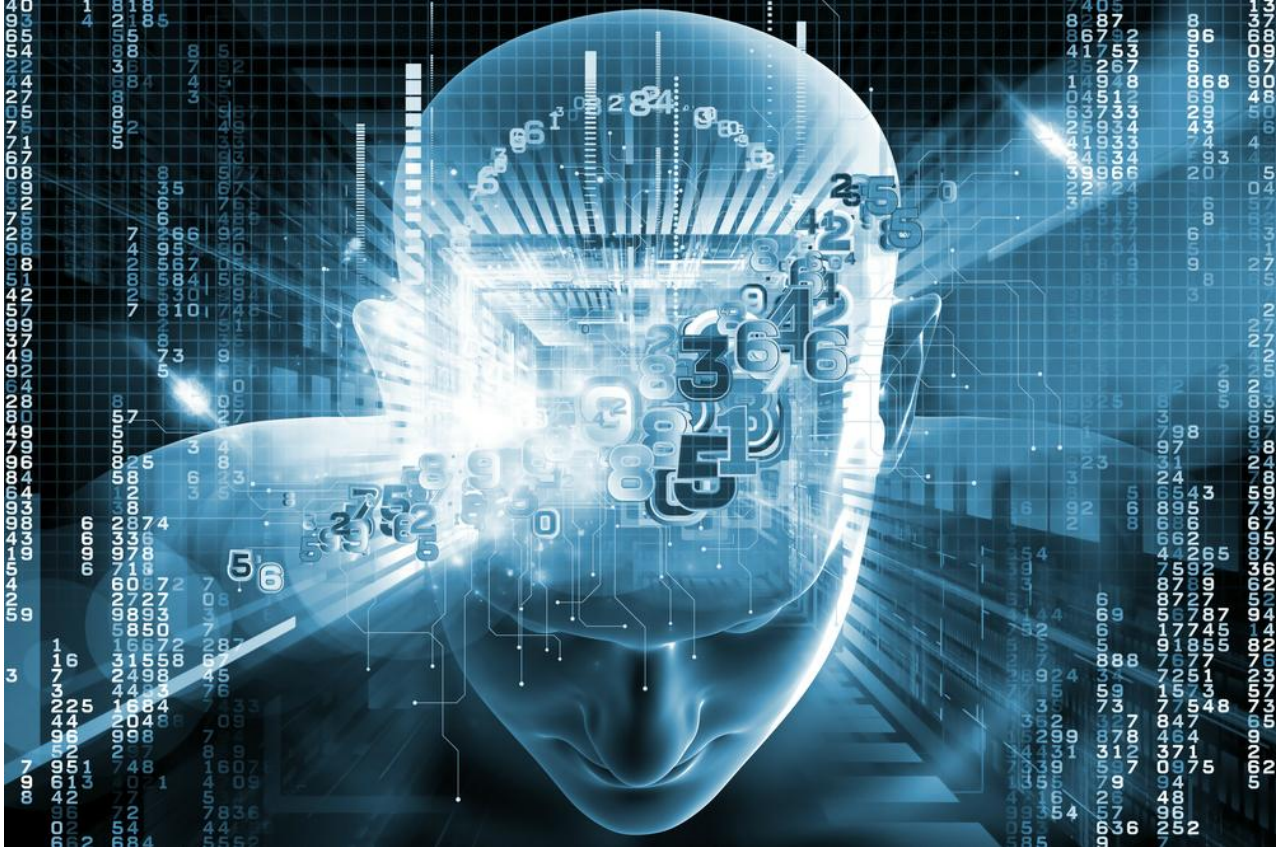
Checking for existence: C:\Users\franklinp\Documents\Battleships\Battleship Stage -1.exe
Executing: C:\Program Files\CodeBlocks\cb_console_runner.exe "C:\Users\franklinp\Documents\Battleships\Battleship Stage -1.exe" (in C:\Users\franklinp\Documents\Battleships)
```

6. After a few rounds

```
mingw32-g++.exe -c "C:\Users\franklinpc\Documents\Battleships\Battleship Stage -1.cpp" -o "C:\Users\franklinpc\Documents\Battleships\Battleship Stage -1.o"
mingw32-g++.exe -o "C:\Users\franklinpc\Documents\Battleships\Battleship Stage -1.exe" "C:\Users\franklinpc\Documents\Battleships\Battleship Stage -1.o"
Process terminated with status 0 (0 minute(s), 0 second(s))
0 error(s), 0 warning(s) (0 minute(s), 0 second(s))
Checking for existence: C:\Users\franklinpc\Documents\Battleships\Battleship Stage -1.exe
Executing: C:\Program Files (x86)\CodeBlocks\cb_console_runner.exe "C:\Users\franklinpc\Documents\Battleships\Battleship Stage -1.exe" (in C:\Users\franklinpc\Documents\Battleships)
```

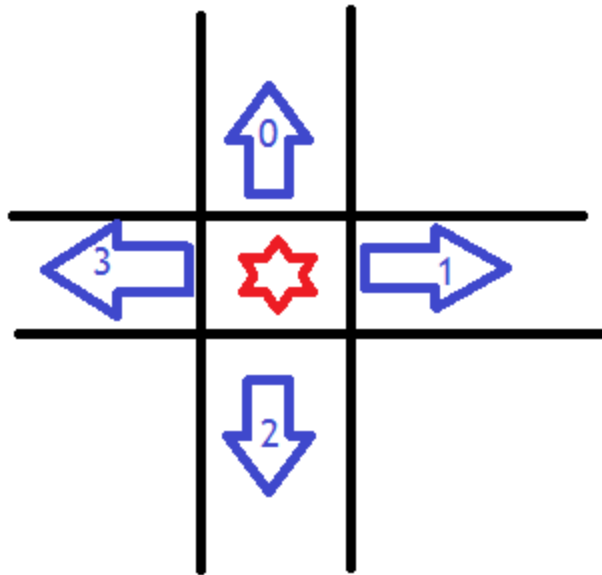
7. here the computer is in a winning position.

The Artificial Intelligence

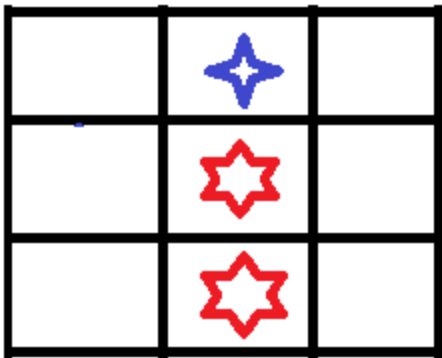


The logic behind the a.i. is that initially the computer does not know where the player's ship is located so he starts by randomly hitting targets. Once he gets lucky and hits a player's ship he stops randomising and goes hitting around the target to get the rest of the ship. The A.I. cannot go randomly hitting targets near the killed block (i will refer kill block as the block killed by the A.I.)so a specific algorithm had to be deployed so not to miss any case.

The a.i. goes around the kill block in a clockwise fashion starting from top. we gave codes to different directions .(red star-killed block, blue star- means hit but miss)



the A.I. goes from 0->1->2->3. On following a direction if it does not find ship i.e by bombing in that direction it changes it direction to the next one . Suppose a case arises where



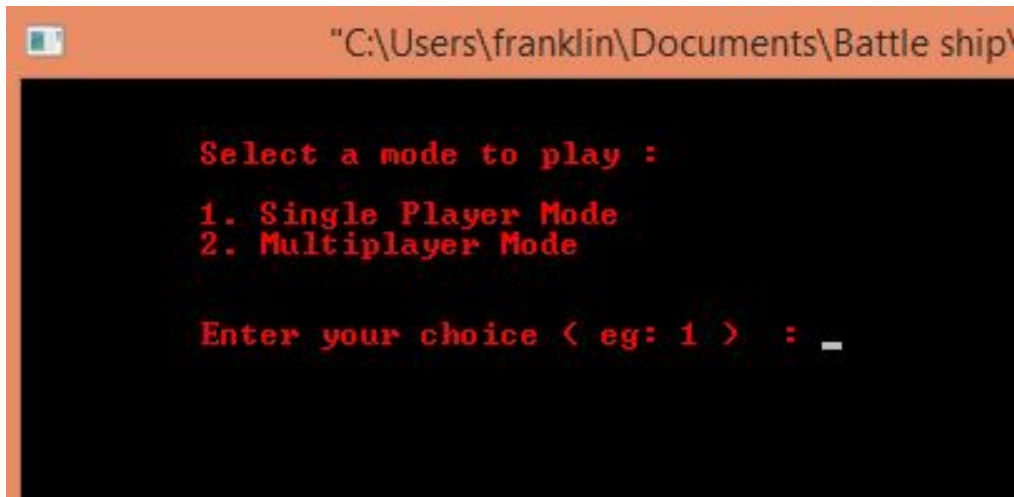
here its obvious that the other part of the ship is in the direction 2 so on encountering such cases we flip the direction to the opposite direction (here 2).

One more maze has been used to save the direction at every position that has to be followed . On hitting successively

we pass on the direction to it i.e the new block so that it can continue with the direction. Care has been taken to avoid blocks were hits have been done and missed ones also the blocks where the computer's ships are located.

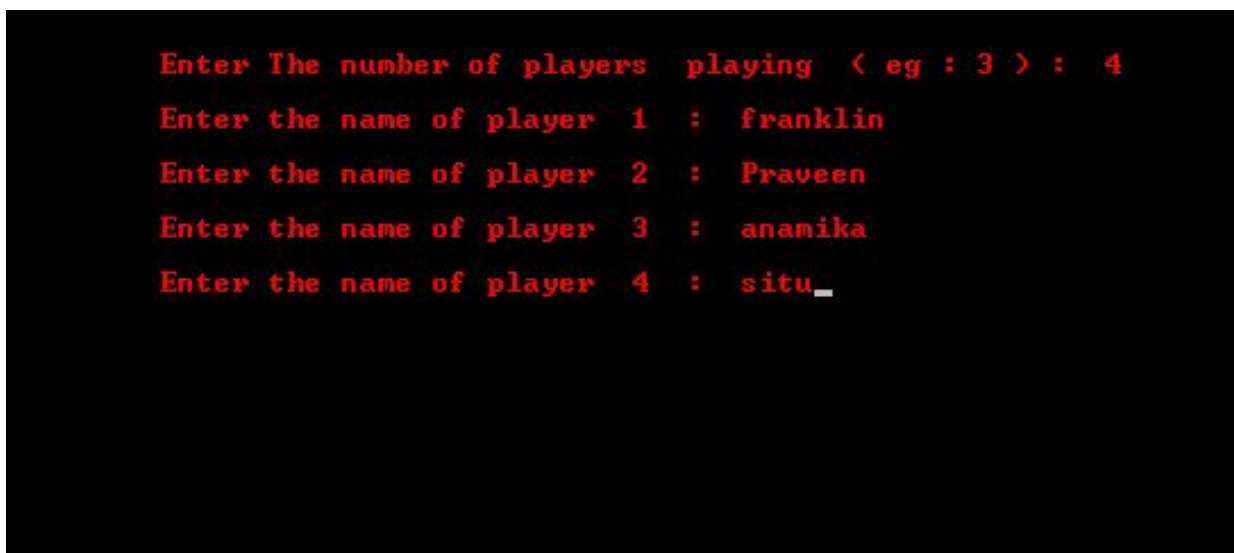
Multiplayer Mode

- The multiplayer mode can be chosen from the menu provided



```
"C:\Users\franklin\Documents\Battle ship\  
  
Select a mode to play :  
1. Single Player Mode  
2. Multiplayer Mode  
  
Enter your choice < eg: 1 > : _
```

- After logging in the admin mode you will be asked to name your players and will be shown the ranks after that.



```
Enter The number of players playing < eg : 3 > : 4  
Enter the name of player 1 : franklin  
Enter the name of player 2 : Praveen  
Enter the name of player 3 : anamika  
Enter the name of player 4 : situ_
```


The Order of Rank is :

1. fra

2. Pra

3. ana

4. sit

Press enter to continue_

- After this you will be asked to place your ship on a 10 X 10 matrix.

Players Maze										Hits									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
0	1									0									
1		1								1									
2			1							2									
3										3									
4										4									
5										5									
6										6									
7										7									
8										8									
9										9									

Please, Enter your choice of co-ordinates for bombing : _

- Now you can chance by chance give inputs in the form of co-ordinates and play the game.
- You can always give input as 10 10 for help.