1. The quickest way to win the game is to destroy the battleship. The algorithm I plan to implement will have two modes. Hunt and Target. In hunt mode the algorithm will fire shots in diagonal lines spaced (Battleship.Length – 1) spaces apart. Such as the fallowing:

Battleship.Length = 7

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R |
| 1 | **X** |  |  |  |  |  | **X** |  |  |  |  |  | **X** |  |  |  |  |  |
| 2 |  |  |  |  |  | **X** |  |  |  |  |  | **X** |  |  |  |  |  | **X** |
| 3 |  |  |  |  | **X** |  |  |  |  |  | **X** |  |  |  |  |  | **X** |  |
| 4 |  |  |  | **X** |  |  |  |  |  | **X** |  |  |  |  |  | **X** |  |  |
| 5 |  |  | **X** |  |  |  |  |  | **X** |  |  |  |  |  | **X** |  |  |  |
| 6 |  | **X** |  |  |  |  |  | **X** |  |  |  |  |  | **X** |  |  |  |  |
| 7 | **X** |  |  |  |  |  | **X** |  |  |  |  |  | **X** |  |  |  |  |  |
| 8 |  |  |  |  |  | **X** |  |  |  |  |  | **X** |  |  |  |  |  | **X** |
| 9 |  |  |  |  | **X** |  |  |  |  |  | **X** |  |  |  |  |  | **X** |  |
| 10 |  |  |  | **X** |  |  |  |  |  | **X** |  |  |  |  |  | **X** |  |  |
| 11 |  |  | **X** |  |  |  |  |  | **X** |  |  |  |  |  | **X** |  |  |  |
| 12 |  | **X** |  |  |  |  |  | **X** |  |  |  |  |  | **X** |  |  |  |  |
| 13 | **X** |  |  |  |  |  | **X** |  |  |  |  |  | **X** |  |  |  |  |  |
| 14 |  |  |  |  |  | **X** |  |  |  |  |  | **X** |  |  |  |  |  | **X** |
| 15 |  |  |  |  | **X** |  |  |  |  |  | **X** |  |  |  |  |  | **X** |  |
| 16 |  |  |  | **X** |  |  |  |  |  | **X** |  |  |  |  |  | **X** |  |  |
| 17 |  |  | **X** |  |  |  |  |  | **X** |  |  |  |  |  | **X** |  |  |  |
| 18 |  | **X** |  |  |  |  |  | **X** |  |  |  |  |  | **X** |  |  |  |  |

Doing it this way assures that I will eventually find the battleship, as there is never a space that is bigger than the length of the battleship. After a hit is made, the algorithm will switch to target mode. After making a hit, it will fire above, below, left and right of each hit. That way it will always destroy any/all ships it finds.

Note: this algorithm will actually work cooperatively with other players to finish off ships. Ill have to add provisions to ignore shots against its own ships.

2.

See FireAlgo.cs. this was written to work with the rest of the code from previous battleship project. It isn’t as detailed as I would like, but its pretty much the outline for it.

3.

shotGrid will be used to hold all registered and internal shots called on any grid. It is a 2d array of chars. PreviousShotPosition is the position that is feed to my algorithm to tell it where all the shots are fired. And PreviousShotResult is the result that is feed to my algorithm.

|  |
| --- |
| FireAlgo.cs |
| internal char[,] shotGrid;  internal Position PreviousShotPosition;  internal String PreviousShotResult;  int SizeOfGrid;  Position TargetLoc;  bool TargetMode = false; |
| RegisterShot(Position shotPos, String Response)  MakeShot() |

5.

The methods for interacting with my algorithm are FireAt(Position shotPos,Fleet CurrentFleet), which is used when I shot is fired against my program. RegisterShot(Position ShotPos, String Response) is used to listen to the responses from the other players, and MakeShot() is executed when its my turn to fire.

All of these can be found in FireAlgo.cs