

Team: Castor Studios

Student1: Dragos Paul Vecerdea (student number 4794117)

Student2: Ionut Laurentiu Contantinescu (student number 4808916)

2.1 Plan of action

Three buttons to choose a type of ship the user wants to place. One button to set horizontal or vertical orientation.

Two tables with cells (my table and enemy table).

Click on a cell in my table to place a ship.

Check if ship is still available to be placed.

Click ready when all the ships have been placed.

Click on the opponents table to attack cells. (Check if cell was attacked before)

Get feedback weather it was a hit or a miss.

2.2

Two matrix mytable and enemytable.

My table will be 0 on mytable[i][j] if row I column j is empty and will have value x if the xth ship was placed on that specific cell.

enemytable[i][j] will tell weather row I column j was attacked before

3 functions: verify, completeover, completeout

Complete over (when the click is hover the table a shadow of the ship to appear on the table)

Complete out (when the click is out of the cell the shadow is removed)

Verify checks if for a specific cell and type of ship it is possible to be placed or not

3.3

There will be a set of message patterns that are sent from the client to the server and from the server to the client.

When the client asks to play the game it sends message PLAY

The server sends DONE if both players are done placing the ships

Sends WAIT if it still searches for an opponent and sends PLACE to ask the users to place the ships.

Sends MOVE to ask a player for a cell to attack.

Sends VERIFY + encoded position of an attacked cell to the opponent

Sends LOST or WIN if the game finished

Sends HIT or MISS to inform the user if the cell he attacked was a hit or a miss

Client sends PC + encoded position to issue an attack on a specific position

Client also sends ILOST when he lost the game