

Naval Battle – Explanation of the realization

The aim of this project is to realize the famous game of naval battle, with the added subtlety of having three depths.

The principle is quite simple: player 1 enters the coordinates (x,y,z) of his three boats (one of one square, one of two squares, and one of three squares), then player 2 does the same for his boats. Then each player takes turns to shoot at a square, hoping to hit an enemy submarine. If a boat is hit, T^1 is displayed where it is hit, if all its components are hit, C^2 is displayed, it is sunk. If no boat is hit but there is a presence on the adjacent squares, V^3 is shown. On the other hand, if no ship is present either on the target square or on the adjacent squares, then R^4 is shown, Nothing to see. The game ends when all three of a player's ships are sunk, so he has lost.

Regarding the storage of variables, we chose to store the data to be displayed in a three-dimensional list, one per sea depth. Regarding the state of the submarines, we stored them in two-dimensional dictionaries: the first one concerns the type of submarine (1, 2 or 3 cells), and the second dimension concerns the coordinates of each part of the submarine. In case of a sudden stop of the game (computer shutdown, etc.), all the variables are stored in a csv file at each shot, and are imported into the program when the main.py file is launched. The only condition is that all the submarines must be installed to make the first save of the game.

1 Touché (Hit)
2 Coulé (Sunk)
3 Vu (Seen)
4 Rien en vue (Nothing to see)

```

Joueur 1 : Vous posez votre sous-marin à 1 cases.
profondeur :0
1ere Coordonnée x :1
1ere Coordonné y :2
Joueur 1 : Vous posez votre sous-marin à 2 cases.
profondeur :2
1ere Coordonnée x :3
1ere Coordonné y :2
2eme Coordonnée x :2
2eme Coordonné y :2
Joueur 1 : Vous posez votre sous-marin à 3 cases.
profondeur :1
1ere Coordonnée x :4
1ere Coordonné y :1
2eme Coordonnée x :3
2eme Coordonné y :1
3eme Coordonnée x :2
3eme Coordonné y :1
Joueur 2 : Vous posez votre sous-marin à 1 cases.
profondeur :0
1ere Coordonnée x :3
1ere Coordonné y :3
Joueur 2 : Vous posez votre sous-marin à 2 cases.
profondeur :2
1ere Coordonnée x :3
1ere Coordonné y :2
2eme Coordonnée x :2
2eme Coordonné y :2
Joueur 2 : Vous posez votre sous-marin à 3 cases.
profondeur :4
1ere Coordonnée x :3
1ere Coordonné y :2
!erreur! Case non existante
profondeur :1
1ere Coordonnée x :4
1ere Coordonné y :3
2eme Coordonnée x :3
2eme Coordonné y :3

```

Player 1 puts its three submarines coordinates

Player 2 puts its three submarines coordinates

Cell doesn't exist

```

3eme Coordonnée x :2
3eme Coordonné y :3
Joueur 2Saisissez la profondeur où vous voulez tirer : 1
Saisissez la coordonnée X où vous voulez tirer : 3
Saisissez la coordonnée Y où vous voulez tirer : 3

```

entry of the coordinates to be targeted

```

0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 R 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 R 0 0 0 0 0 0
0 0 R R R 0 0 0 0 0
0 0 0 R 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 R 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0

```

display of the targeted sea