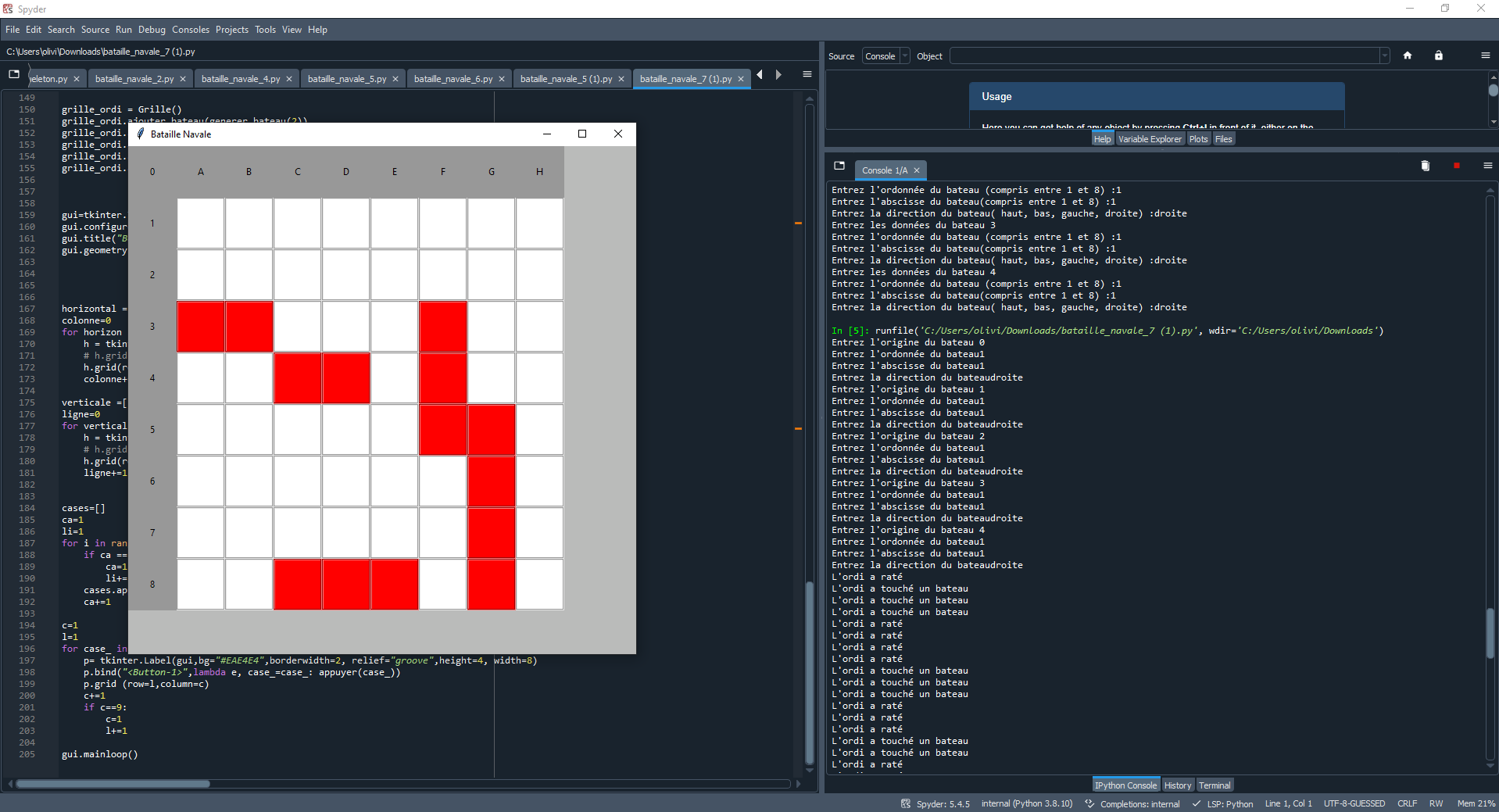
|  |  |  |
| --- | --- | --- |
|  | **BATAILLE NAVALE**  **PYTHON** | **Elric Dufresne**  **Thomas Jenvrin** |

**Description générale :**

La Bataille navale oppose 2 joueurs sur une grille carré de 10 par 10.

Le premier joueur doit positionner 5 bateaux dans cette grille, puis l’ordinateur fait de même.

Alternativement chacun tente de couler le bateau de l’autre en donnant les coordonnées du tir.



**Cahier de charges :**

Le jeu doit permettre à un joueur de placer 5 bateaux sur la grille.

Le programme doit générer le placement pour les 5 bateaux adverses.

Le joueur doit pouvoir saisir les coordonnées de ses tirs.

Le programme doit vérifier si le tir est raté ou touché et s’il reste des bateaux à couler.

L’interface de saisie doit être simple et accessible.

La restitution des résultats doit être claire et explicite.

**Choix des outils :**

On utilise le langage Python 3 et les affichages se feront avec la bibliothèque tkinter

**Structure du programme :**

**Nous avons 2 Objets : Grille et Bateau**

A. OBJETS :

1. Grille :

* 1.1. Attributs (initialisation de la grille) :
* 1.1.1. Grille.bateaux :
* C'est un tableau qui contient des objets de type Bateau, qui représentent les bateaux posés sur la grille du joueur.
* 1.1.2. Grille.casesTestees :
* C'est un tableau qui contient des tableaux contentant toutes les cases qui ont été testées avec leur résultat (si le bateau a été touché ou pas)
* 1.1.3. Grille.casesTouchees :
* Cette variable contientn le nombre de cases touchées.
* 1.2. Méthodes :
* 1.2.1. Grille.valide()
* vérifie si les bateaux sont bien positionnés. Renvoie True si tous les bateaux sont bien dans la grille, valides et au nombre exact de 5 ; il renvoie False si les bateaux sont mal positionnés, incorrects ou qu'il n'y en a pas 5.
* 1.2.2. Grille.case(coordonnees) :
* Cette méthode parcourt tous les bateaux un par un et vérifie si il y a un bateau qui contient cette case. Elle renvoie True si un bateau est sur cette case, et renvoie False si aucun bateau n'est sur cette case (raté).
* Elle renvoie False si la case est incorrecte (par exemple, si l'utilisateur entre (Z;72)).
* Cette méthode ajoute dans l'attribut de type Tableau un élément qui contient mes coordonnées entrées par l'utilisateur et un booléen qui indique si un bateau a été touché.
* 1.2.3. Grille.bateaux()
* Cette méthode renvoie les bateaux présents sur la grille.

2. Bateau :

* 2.1. Attributs (initialisation du bateau) :
* 2.1.1. coordonnées : c'est un tuple qui contient les coordonnées de l'origine du bateau.
* 2.1.2. direction : il s'agit d'une chaîne de caractères qui indique la direction du bateau (haut, bas, droite, gauche)
* 2.1.3. longueur : cette variable contient la longueur du bateau sous forme d'un entier compris entre 2 et 5 inclus.
* 2.1.4. casesTouchees : c'est un tableau qui contient toutes les cases qui ont été touchées.
* 2.1.5. \_\_cases\_trouvees : nombre de cases trouvées.
* 2.1.6. \_\_coule : booléen qui indique False si le bateau n'est pas coulé, True dans le cas contraire.
* 2.1.7. Bateau.Cases :
* Il s'agit d'une liste qui contient toutes les cases du bateau, sous forme de tuples.
* 2.2. Méthodes :
* 2.2.1. Bateau.creer(l, origine, direction) :
* Il crée le bateau de longueur l, en partant de l'origine (x,y) dans la direction indiquée (haut, bas, gauche, droite)

**Répartitions des taches :**

Thomas :

* objet grille
* programme principal

Elric :

* objet bateau
* tkinter

**Difficultés :**

****

**Ouvertures possibles :**

Ajouter un espace d’accueil pour lancer le jeu, avec les règles

Ajouter un affichage sur une grille des cases testées.

Ajouter un affichage de fin