Avoir Python d'installer et créer une variable d'environement.

installer Tkinter :

Depuis le CMD : pip install tkinter

Lancer le Bataille.bat

Detail du code :

1. Classe Navire

- **But** : Représenter un navire, avec un nom (ex : "Porte-avions"), une taille (ex : 5), la liste de ses positions sur la grille (positions) et les positions touchées (positions_touchees).
- **Attributs** :
    - nom (str) : nom du navire (ex. "Croiseur").
    - taille (int) : nombre de cases occupées (ex. 4).
    - positions (list[tuple[int, int]]) : coordonnées (row, col) où est placé le navire.
    - positions_touchees (list[tuple[int, int]]) : sous-ensemble de positions indiquant quelles cases ont été touchées.
- **Méthodes** :
    - est_coule() : retourne un booléen indiquant si le navire a été entièrement touché.

2. Classe Joueur

- **But** : Représenter un joueur (humain ou ordinateur).
- **Attributs** :
    - nom (str) : nom du joueur.
    - navires (list[Navire]) : liste des navires qui lui appartiennent.
    - grille (list[list[Navire|None]]) : grille 10x10 contenant None (si vide) ou un navire.
    - tirs_effectues (set[tuple[int, int]]) : positions déjà ciblées par ce joueur.
    - mode_difficile (bool) : si True (pour l'IA), l'ordinateur tente de tirer autour des dernières touches.

- reserve_cibles_proches (list[tuple[int, int]]) : cases adjacentes à tirer (pour l'IA en mode difficile).
- tirs_reussis (int) : nombre de tirs qui ont touché ou coulé un navire.
- tirs_rates (int) : nombre de tirs qui ont manqué.

- **Méthodes** :
  - toggle_mode_difficile() : bascule entre mode facile et difficile.
  - initialiser_navires() : crée tous les navires définis dans Navire.NAVIRES_DISPONIBLES.
  - navires_non_places() : retourne la liste des navires qui ne sont pas encore placés sur la grille.
  - get_navire_by_name(name) : retourne l'objet Navire portant le nom name.
  - peut_placer_navire(navire, start_row, start_col, orientation) : vérifie si un navire peut être placé en start_row, start_col (horizontal ou vertical) sans dépasser la grille ou chevaucher un autre navire.
  - placer_navire(navire, start_row, start_col, orientation) : place effectivement le navire sur la grille (met à jour navire.positions et self.grille).
  - placement_aleatoire() : place tous les navires de façon aléatoire (pour l'ordinateur).
  - tirer_sur(autre_joueur, row, col) : effectue un tir sur la grille de autre_joueur aux coordonnées (row, col), et retourne le résultat ("touche", "coule", "manque", "deja_tire"). Met à jour les compteurs tirs_reussis / tirs_rates.
  - tous_navires_coules() : indique si tous les navires du joueur sont coulés.

3. Classe Plateau

- **But** : Gérer l'affichage d'un plateau (un canevas Tkinter), ainsi que la lecture des sons correspondants.
- **Attributs** :
  - parent (Tk ou Frame) : parent Tkinter.
  - rows, cols (int) : dimensions du plateau (par défaut 10x10).
  - cell_size (int) : taille en pixels d'une cellule (par défaut 30).
  - canvas (Canvas) : canevas pour dessiner la grille et colorier les cases.

- grid_color (list[list[str]]) : mémorise la couleur permanente de chaque case.
- preview_items (list[int]) : mémorise les ID graphiques des rectangles temporaires de prévisualisation.
- joueur (Joueur) et ordinateur (Joueur) : dans cet exemple, on instancie deux joueurs, mais on peut adapter selon le besoin.
- orientation_joueur (StringVar) : stocke l'orientation ("H" ou "V") pour le placement.
- phase (StringVar) : stocke la phase de jeu ("placement", "battle", "fin").

- **Méthodes** :
  - draw_grid() : dessine le quadrillage de base.
  - color_cell(row, col, color) : colorie de façon permanente la case (row, col) en color.
  - clear_preview() : efface les rectangles de prévisualisation.
  - color_preview_cell(row, col, color) : dessine temporairement un rectangle coloré sur la case (row, col).
  - redraw_all_cells() : redessine toutes les cases permanentes (par exemple après une réinitialisation).
  - toggle_orientation() : bascule l'orientation de placement entre "H" et "V".
  - play_sound_tir(), play_sound_touche(), play_sound_coule() : jouent les sons correspondants à l'action effectuée.

4. Fonction main()

- **But** : Point d'entrée du programme.
- **Rôle** :
  1. Crée la fenêtre Tkinter.
  2. Instancie les objets Joueur (humain, ordinateur).
  3. Met en place les interfaces graphiques (plateaux, boutons, labels).
  4. Gère la boucle d'événements Tkinter (clic souris, phase de placement, passage en phase de combat, etc.).