Battleship game project

Objective:

- The objective of the project is to create a battleship game. It is meant to be a single-player game, where player would face a computer controlled enemy.

Description:

- 10x10 board consists of rows 1-10 and columns A-J
- Squares are represented by symbols:
  - ` ` - sea
  - ~ - attacked sea
  - O - ship
  - X - attacked ship
- Player, like enemy has 10 ships:
  - 4x 1-square
  - 3x 2-square
  - 2x 3-square
  - 1x 4-square
- Ships can be placed vertically or horizontally
- Ships can't be placed near or on other ship
- Game is played in turns
- Every turn player and computer can guess one square
- When whole ship is found squares around it become attacked sea
- Game ends when either player or enemy has no ships left

Project structure:

Project's main file is battleship_game.py. It uses curses module for displaying the game , time and Enum module, as well as Ship, BattleshipBoard, Player and ComputerPlayer classes:

- Ship:

  Main purpose of this class is to allow representing ships with first position, direction and length. Thanks to positions method it is possible to convert this data into list of positions that ship occupies. It also has move_ship and rotate_ship methods that allow for the player to position his ships.

- BattleshipBoard:

  Its purpose is to store game board and allow to change it with methods like: set_square, set_square, place_ship, set_squares_around_ship. One useful method is get_ship_if_whole_sunk. It detects if near or in given position there is a sunk ship and returns instance of that ship if found one. Also object has it's string representation.

- Player:

  It stores player's board and player's guess board. Also it allows player to make guesses and has method that can place ship in random and not occupied squares.

- ComputerPlayer:

  It stores computer board. Like Player class it has place_ship and guess methods. However, guess method is different from player's. In stead of checking given position it chooses random not checked one and if ship was found searches for other ship squares until whole ship is found.

Things like length and amount of ships, square indications and board size can be changed in game_settings.py in game_files directory, but it's not guaranteed that game will function properly, since it was created with intention of playing on settings present in this file.

User guide:

To play the game you should download project from:

https://gitlab-stud.elka.pw.edu.pl/plenczew/battleship_game_project/-/tree/main and open whole directory in terminal. Then use command:

```
python battleship_game.py
```

Make sure that terminal window is big enough to display game, otherwise it won't start. Note that project uses curses, Enum and time modules, which are required for game to function.

When game starts act accordingly to messages under title. If you don't want to finish game you can close it with Ctrl + C.

First positioning phase starts. You can move ship with arrows, rotate with v, or place it with x. New ships will pop up until you place all 10 ships. Then Guess phase starts. To guess square all you have to do is type square name. It consists of first column A-J and row 1-10 (for example A3). When you have no ships you lose, when you guess all ships you win. When game is over click any key to exit.

Authors' thoughts:

I've been working on this project for few months. Since the beginning a few changes have occurred. These were my original goals:

- Making class that would allow me to type first position, direction and length of ship and convert it to its positions
- Creating my own battleship board that would store squares with double list and allow me to get individual square by giving position of this square
- Creating a dictionary, that would translate square name to position
- Creating a way for player to guess and place ships in board

- Doing the same thing for computer player but it would search for ship squares if it found one
- Creating a string representation of a battleship board
- Creating a file that would allow me to run the game in terminal, then potentially in GUI

As I've been working, few ideas formed or were given to me:

- Making squares around ship, that was found whole, attacked sea, because there is no possibility for ship to be there. First I wanted to implement it only for computer as it was much easier
- Using curses module to create visual side of the game in terminal
- Using Enum to represent keys, directions and courses
- Changing the way that ships were placed from typing ship attributes, to using keys to move them
- Translating square name directly in BattleshipBoard class
- Simplifying the code

At the end I think I reached most of my goals. What wasn't done is

- making GUI version of game, because I wanted to wait with it till laboratory classes with GUI, but what was shown wasn't really suited for making games and I didn't have enough time to learn how to make GUI for games from scratch. Also, I just got interested with curses module and liked simple design I got.
- translating square names in BattleshipBoard, because it would require too much changes and would complicate the code
- Simplifying the code, I think I managed to achieve it partially as my code was much more complicated, but I am still not satisfied. I just ran out of time even thought I've put many hours into to it.