PyNeutron implementation docs

The game is subdivided into four modules: `main`, `neutron`, `player` and `util`.

main module

This module contains no classes, and instead serves as an entry point to the game. Its tasks consist of setting up `ArgumentParser` instance, and constructing the game based on the parsed arguments from command line.

neutron module

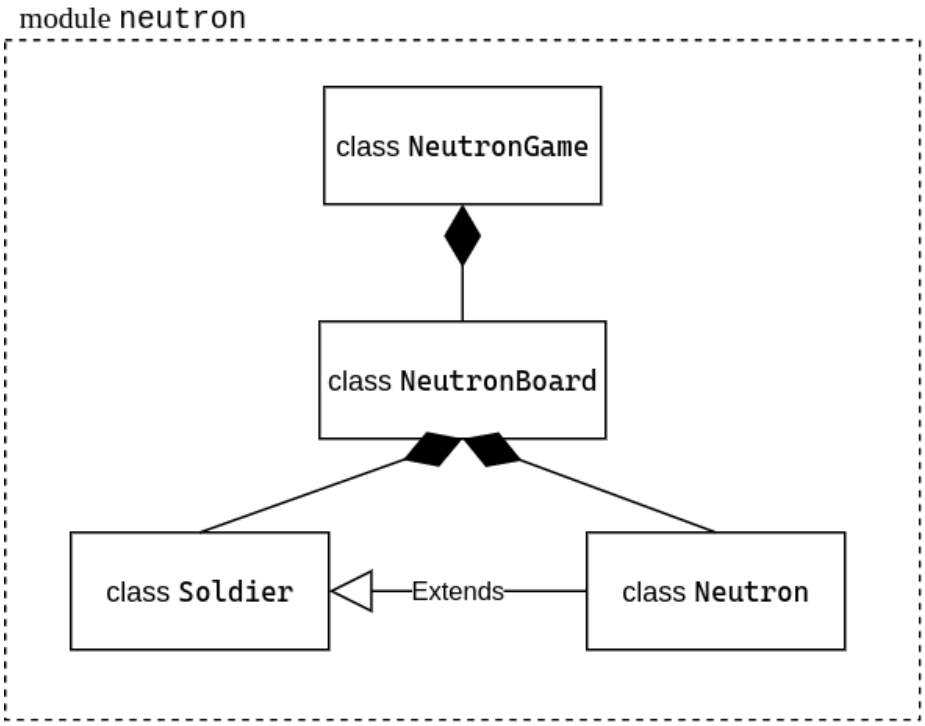

Figure 1: `neutron` module class diagram

This module contains the core logic of the game. `NeutronGame` class is responsible for managing turns, calling players to execute their moves, checking the winning conditions, and terminating the process once the are met. `NeutronBoard` manages the game board, keeps lists of `Soldier` and `Neutron` objects and contains utility functions to query the state of the board. Finally, `Soldier` and `Neutron` objects are used to encapsulate operations on the board in a safe way, to prevent players putting the board in an invalid state.

player module

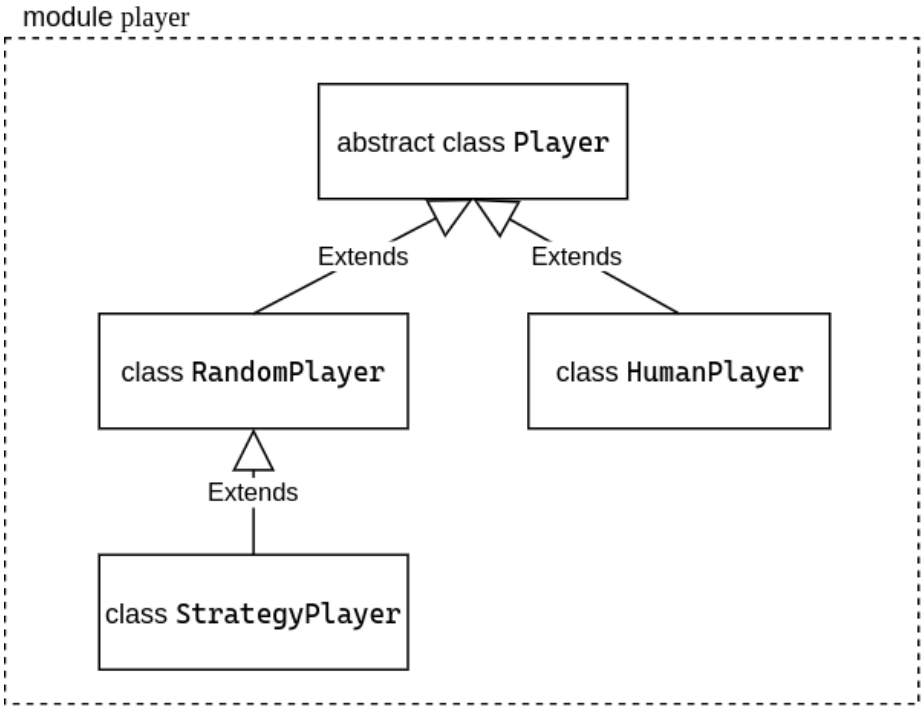

Figure 2: `player` module class diagram

This module houses a hierarchy of `Player`s, which are objects that encapsulate decision-making methods used to move soldiers on the board. They can range from procedures getting data from the user, to sophisticated machine learning algorithms, though the ones included here are not of the latter category. The `Player` class is an abstract base class being the root of this hierarchy. `RandomPlayer` is a player that chooses its moves randomly, `StrategyPlayer` tries to apply some strategies to the moves, but if no strategy can be chosen in the current situation, it reverts to moving randomly, and `HumanPlayer` gets its input from user and moves the soldiers accordingly.

util module

Miscellaneous utilites useful in implementing the game's internals. They include the `Vec` class, which is a simple 2D vector used to operate on positions, `Color` class which is an enumeration of colors used in the game, and dictionaries containing direction names and their abbreviations.

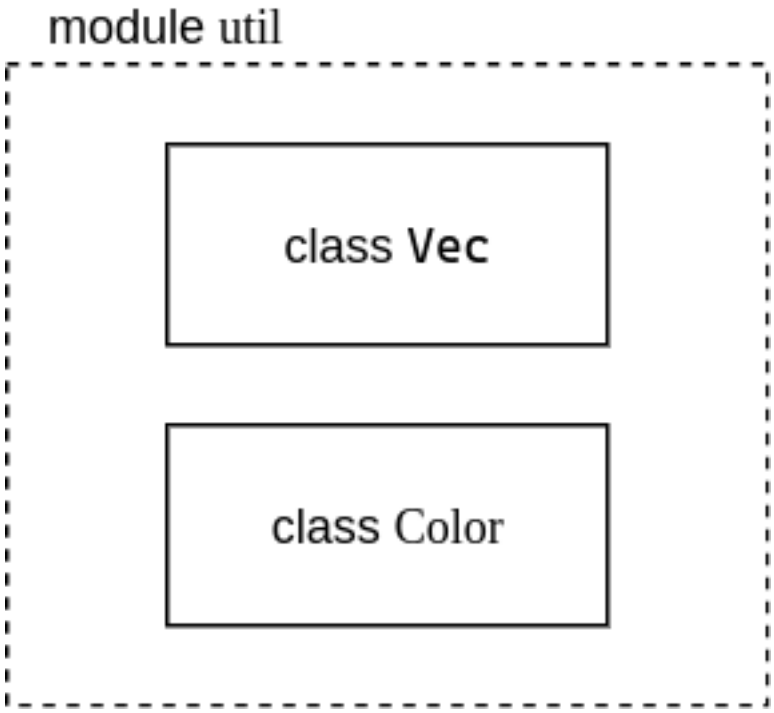

Figure 3: `util` module class diagram