Python Paradigm

When developing my Python solution I used an imperative paradigm. This is easily distinguishable by the lack of classes and functions. Imperative focuses heavily on the sequence of statements as well as the order in which they are executed. If any of the statements within the code are in an incorrect order it can affect the quality as well as the performance of the code. When developing my specific solution I ensured that all statements were in the correct order to ensure none of the logic of the program was disrupted. This involved creating the board and every single instance of a boat first. Imperative languages also have great emphasis on iteration, this meant each boat had to be individually created one by one and used a series of while and for loops within the generation. Once the board and all the boats were generated, I could implement the user interaction. This meant using a series of for and while loops again in order to constantly check the current status of the game, to ensure that the game ended, when necessary, otherwise errors would occur. There was a constant comparison on how many tiles were successfully hit and how many lives the player had, as soon as each were over the program ended. During these loops the program also had to take each user input, validate it with loops and then also compare the players guess to any used tiles on the board. The order of all of this code was extremely important in order to ensure that the program ran smoothly.