Design Patterns

In our Project 3 implementation of a Wordle game, the design pattern most prevalent was a behavioral one, namely the chain of responsibility design pattern. Frankly, our project did not require a very sophisticated design pattern because the objective of the game was simple. This was because in our implementation, we created multiple functions to create different objects and allowed them to interact with the arrays used to store the user’s inputs as well as the functions in charge of checking to see if their input was correct. We created different functions to oversee tasks that did not directly interact with the loop used to keep the interface of the game running. The functions responsible for creating the six grids used to keep track of the guessed words and the alphabet are called prior to the looping condition of the game. After the objects are created, the interface then calls one function that is responsible for calling other helper methods to allow the user to give a guess for each of the six turns. The program will then append the users input to the array for row 1 and then check to see if that word is the same or shares any letters with the word of the day. The chain of responsibility just continuously gets passed from the first row function to the last row function, appending the users guesses into the guess list, while also indicating to the user if any of the letters are in the word and/or in the right position. The program will halt if the user guesses the correct word or runs out of turns.