Functional programming aspects

- **(mostly) side effect free functions**

```
def update_progress(self, letter, indexes):
    for index in indexes:
        self.game_progress[index] = letter
```

```
def updating_scores(self, user_name, wrong_guess):
    x = lambda wrong, score: score - (10*wrong)
    if wrong_guess != 0:
        self.total_score = x(wrong_guess, self.total_score)

    text_file = open(self.file_path, "a")
    text_file.write("\n{0}".format(user_name))
    text_file.write(" {0} ".format(str(self.total_score)))
    text_file.close()
```

- **the use of higher-order functions**

```
def find_indexes(self, letter):
    return [i for i, char in enumerate(self.secret_word) if letter == char]

def is_invalid_letter(self, input_):
    return input_.isdigit() or (self.regex.search(input_) is not None) or (input_.isalpha() and len(input_) > 1)
```

- **use anonymous functions**

```
def updating_scores(self, user_name, wrong_guess):
    x = lambda wrong, score: score - (10*wrong)
    if wrong_guess != 0:
        self.total_score = x(wrong_guess, self.total_score)

    text_file = open(self.file_path, "a")
    text_file.write("\n{0}".format(user_name))
    text_file.write(" {0} ".format(str(self.total_score)))
    text_file.close()
```