Group Assignment Report

Members:

Student ID	Name
814994	Amrit Singh
815290	Karamveer Kaur
815203	Saanch

Objective: A Python program to create a word sorting game. This program allows users to sort a list of random shuffled words.

Program Overview: This word-sorting game has been implemented in Python using various programming constructs such as classes, methods, loops, sorting algorithms and conditions. The program includes functionalities to shuffle the letters of a word, display the scrambled word in a box format, and compare the user's guess to the original word.

Code Implementation: The word sorting game includes a WordSortingGame class with several methods, such as shuffle_word, random_chooser, play_round, and play_game.

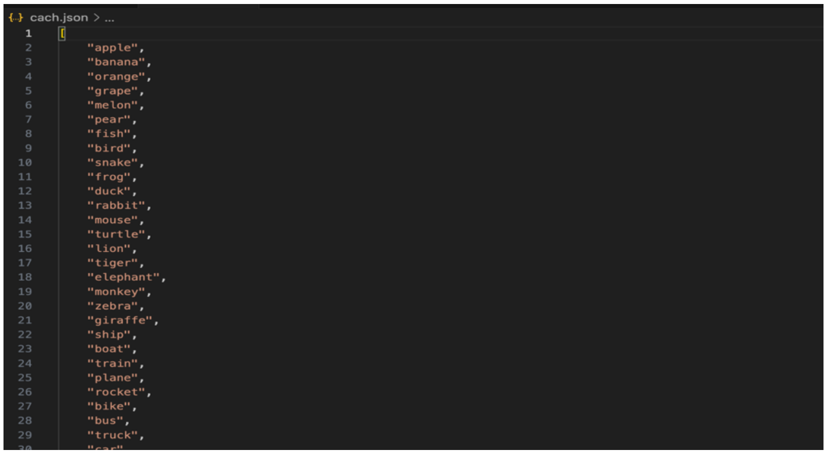
shuffle_word: This method shuffles the letters of a given word using sorting method and displays it in a box format.

randam_chooser: This method passes a random word from the word list.

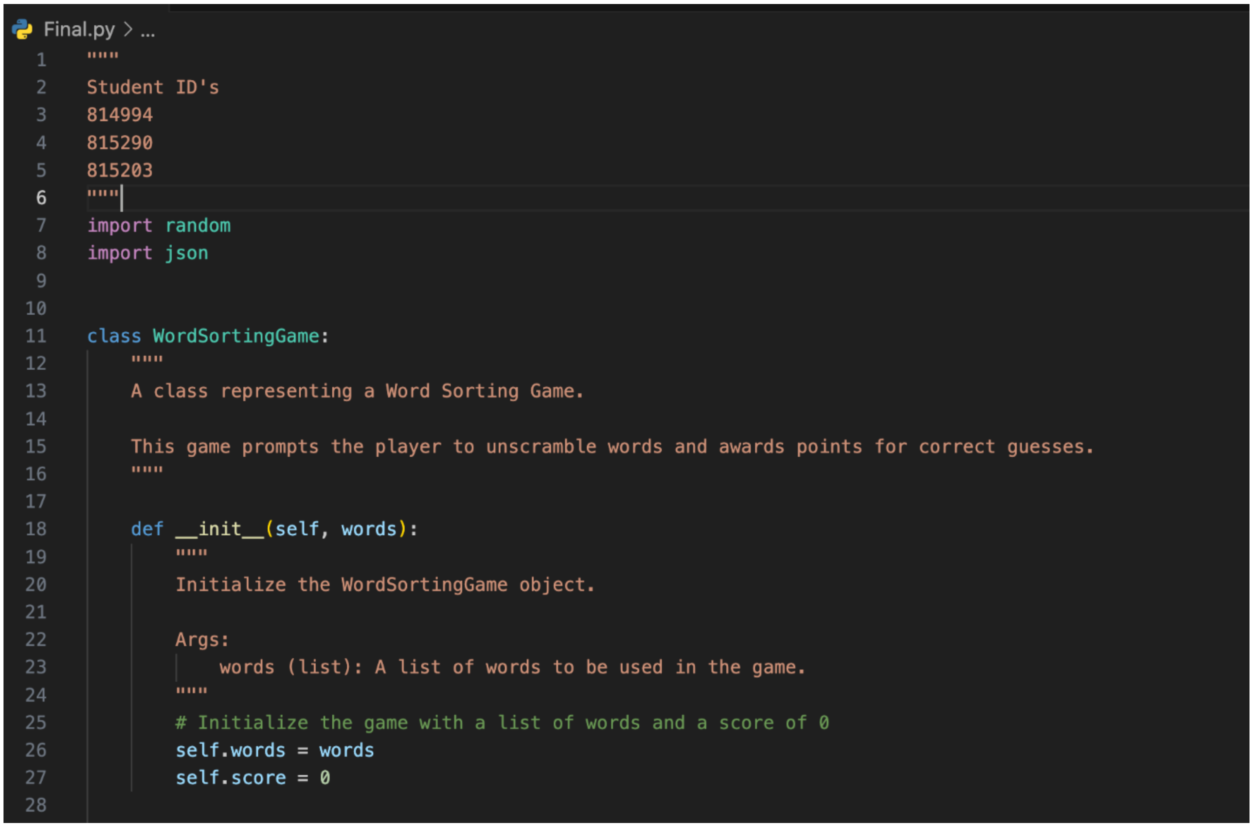
play_round: This method plays a single round of the game, prompting the user to guess the original word from a shuffled word.

play_game: This method manages the entire game, allowing users to play multiple rounds until they choose to quit.

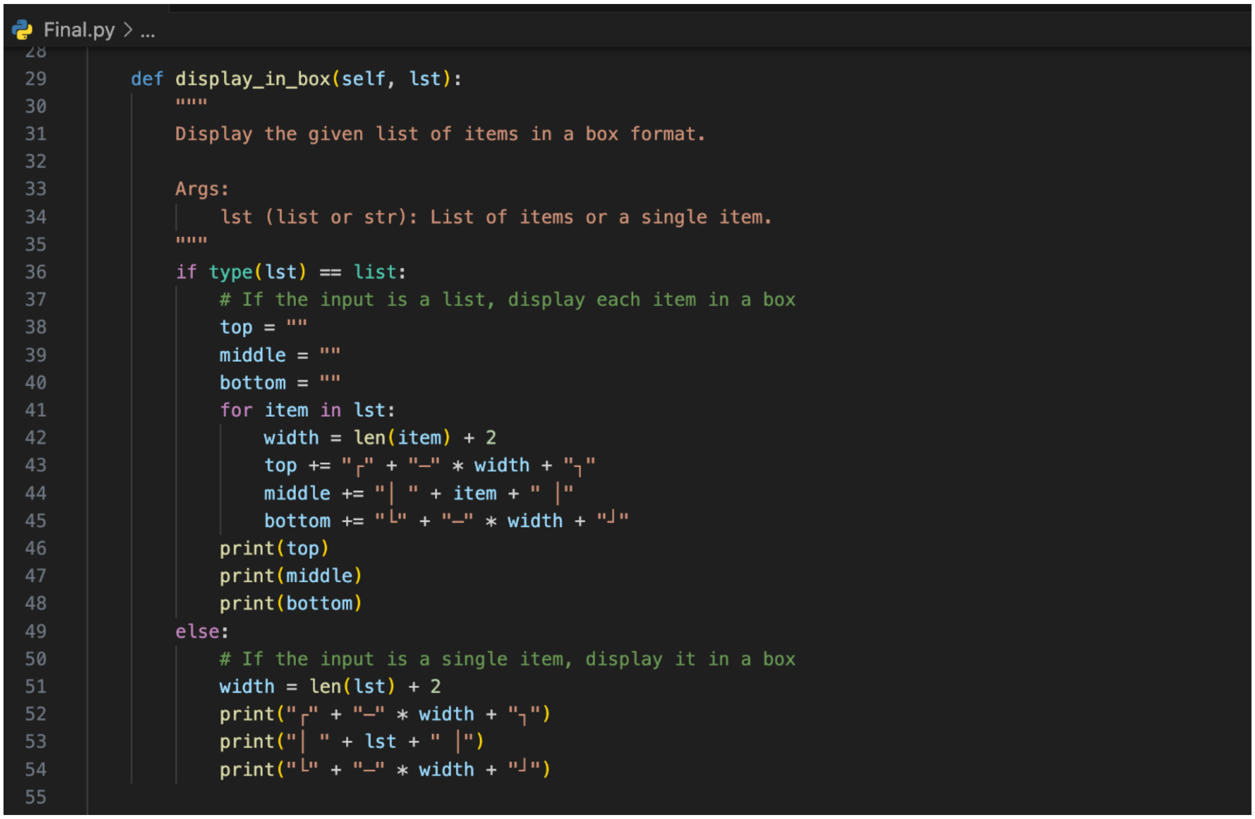
Screenshots of Code implication: -



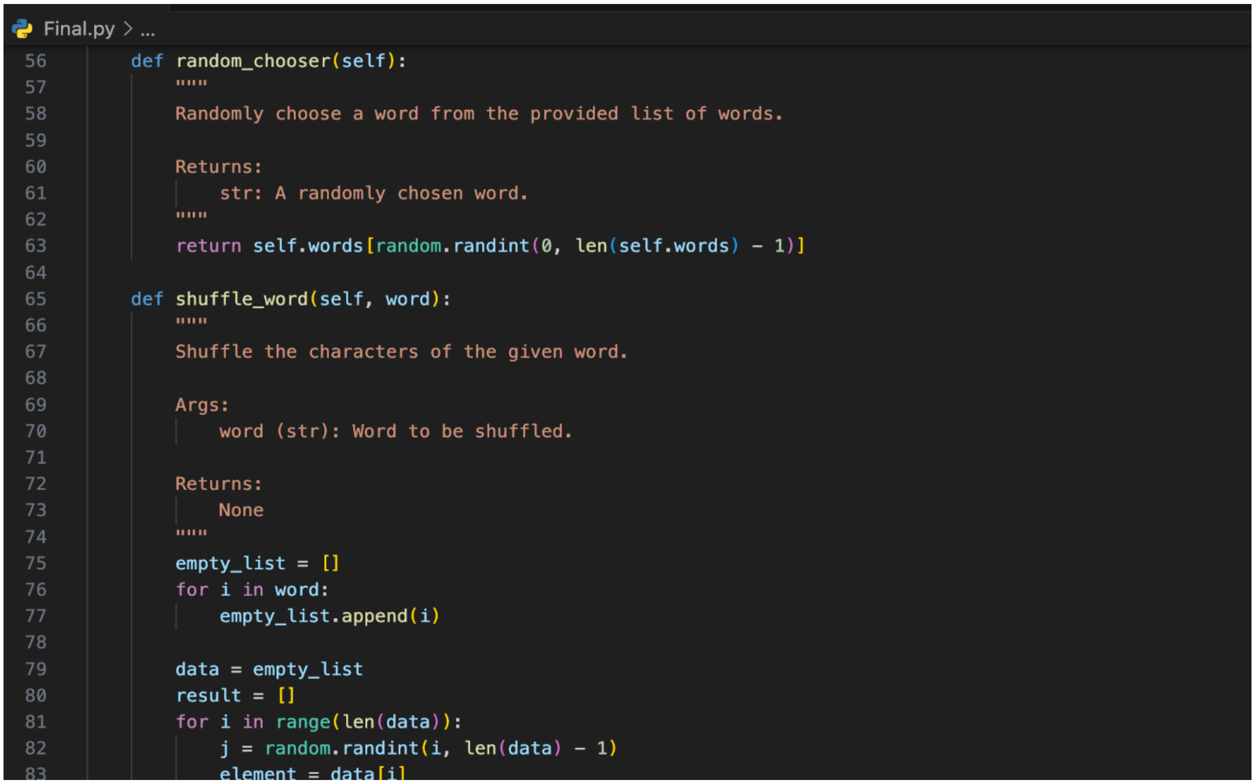
```
[
    "apple",
    "banana",
    "orange",
    "grape",
    "melon",
    "pear",
    "fish",
    "bird",
    "snake",
    "frog",
    "duck",
    "rabbit",
    "mouse",
    "turtle",
    "lion",
    "tiger",
    "elephant",
    "monkey",
    "zebra",
    "giraffe",
    "ship",
    "boat",
    "train",
    "plane",
    "rocket",
    "bike",
    "bus",
    "truck",
    "car"
```



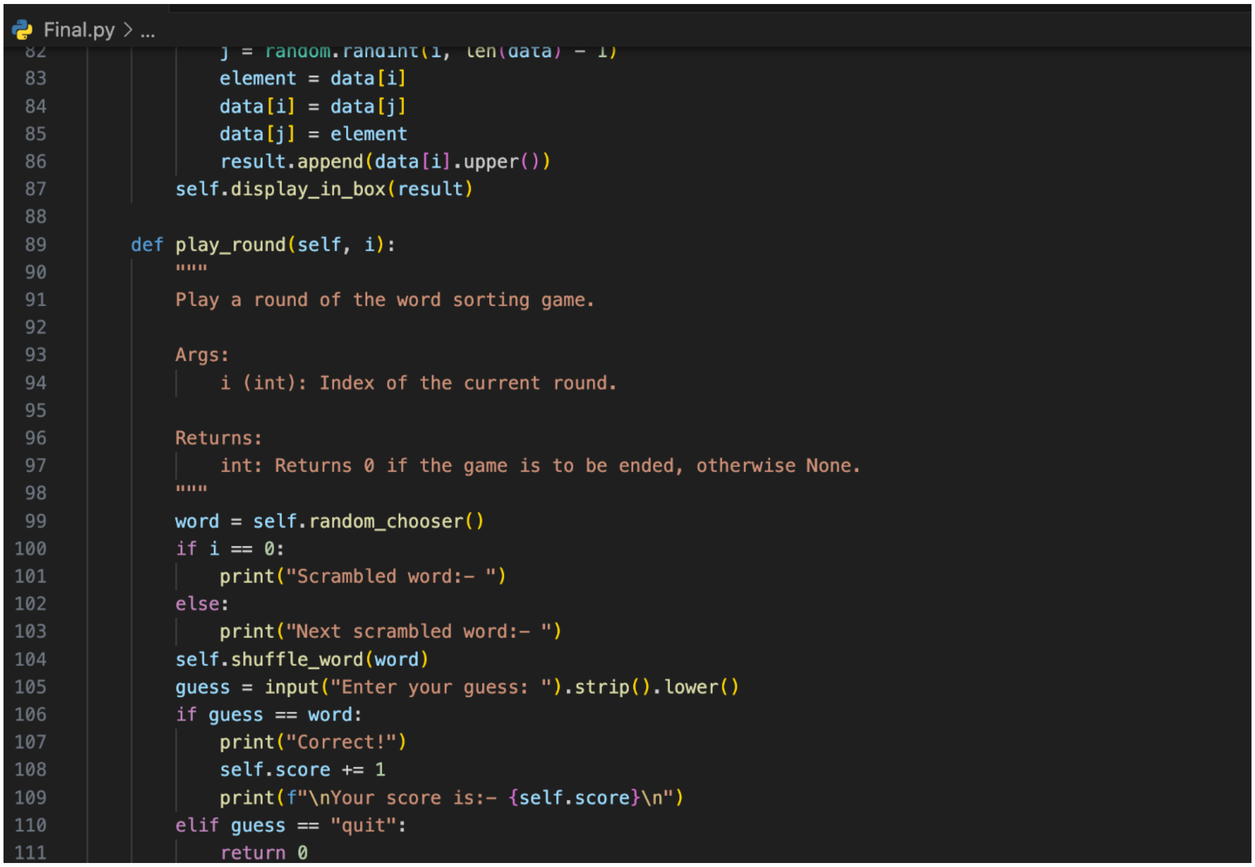
```python
"""
Student ID's
814994
815290
815203
"""
import random
import json


class WordSortingGame:
    """
    A class representing a Word Sorting Game.

    This game prompts the player to unscramble words and awards points for correct guesses.
    """

    def __init__(self, words):
        """
        Initialize the WordSortingGame object.

        Args:
            words (list): A list of words to be used in the game.
        """
        # Initialize the game with a list of words and a score of 0
        self.words = words
        self.score = 0

```



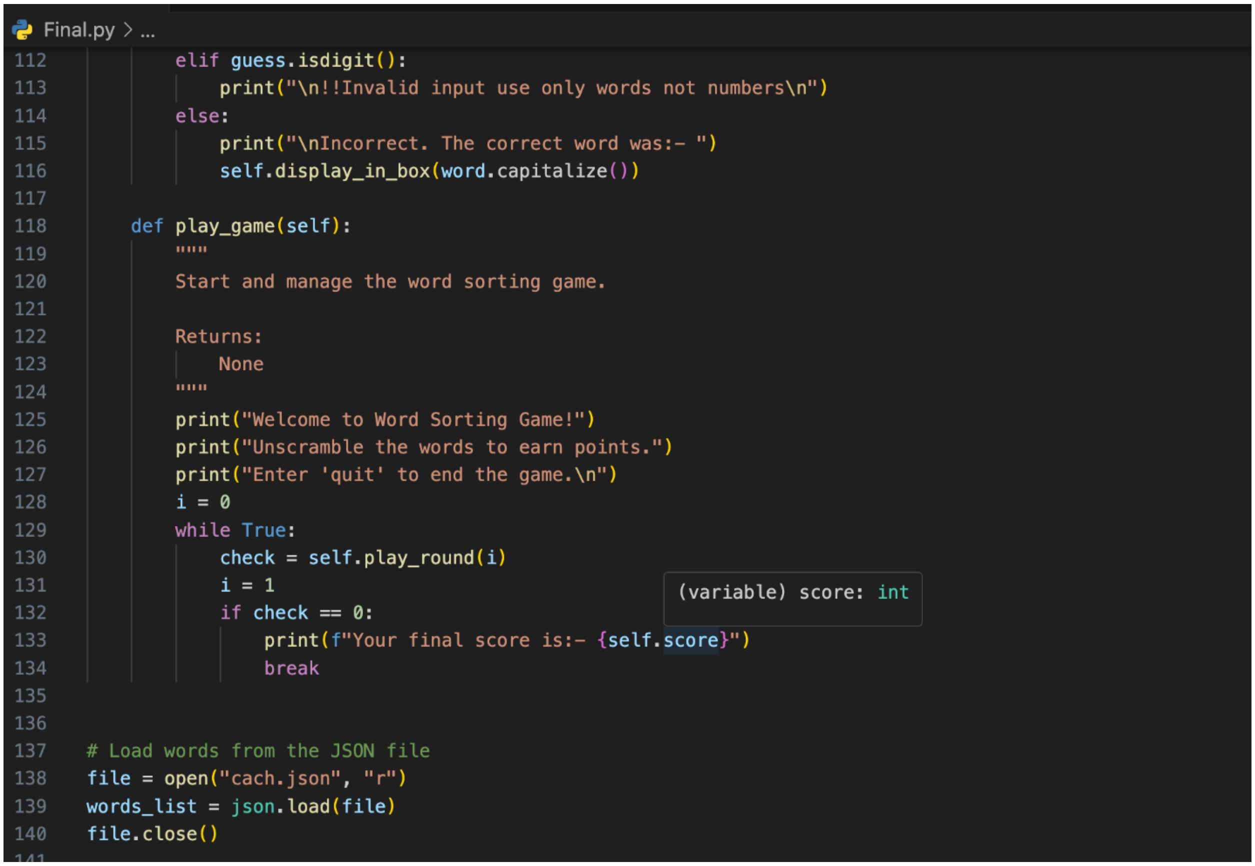
```python
    def display_in_box(self, lst):
        """
        Display the given list of items in a box format.

        Args:
            lst (list or str): List of items or a single item.
        """
        if type(lst) == list:
            # If the input is a list, display each item in a box
            top = ""
            middle = ""
            bottom = ""
            for item in lst:
                width = len(item) + 2
                top += "┌" + "─" * width + "┐"
                middle += "│ " + item + " │"
                bottom += "└" + "─" * width + "┘"
            print(top)
            print(middle)
            print(bottom)
        else:
            # If the input is a single item, display it in a box
            width = len(lst) + 2
            print("┌" + "─" * width + "┐")
            print("│ " + lst + " │")
            print("└" + "─" * width + "┘")

```



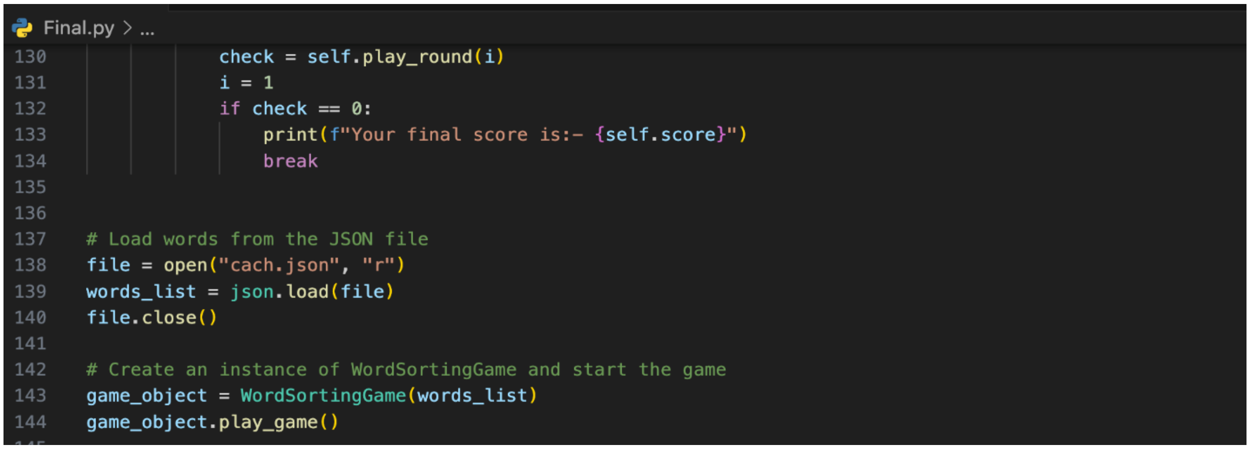
```python
    def random_chooser(self):
        """
        Randomly choose a word from the provided list of words.

        Returns:
            str: A randomly chosen word.
        """
        return self.words[random.randint(0, len(self.words) - 1)]

    def shuffle_word(self, word):
        """
        Shuffle the characters of the given word.

        Args:
            word (str): Word to be shuffled.

        Returns:
            None
        """
        empty_list = []
        for i in word:
            empty_list.append(i)

        data = empty_list
        result = []
        for i in range(len(data)):
            j = random.randint(i, len(data) - 1)
            element = data[i]
```



```python
            j = random.randint(i, len(data) - 1)
            element = data[i]
            data[i] = data[j]
            data[j] = element
            result.append(data[i].upper())
        self.display_in_box(result)

    def play_round(self, i):
        """
        Play a round of the word sorting game.

        Args:
            i (int): Index of the current round.

        Returns:
            int: Returns 0 if the game is to be ended, otherwise None.
        """
        word = self.random_chooser()
        if i == 0:
            print("Scrambled word:- ")
        else:
            print("Next scrambled word:- ")
        self.shuffle_word(word)
        guess = input("Enter your guess: ").strip().lower()
        if guess == word:
            print("Correct!")
            self.score += 1
            print(f"\nYour score is:- {self.score}\n")
        elif guess == "quit":
            return 0
```

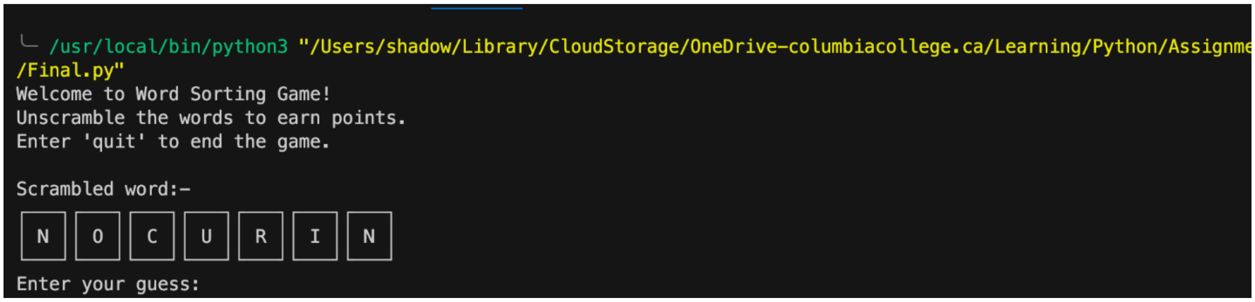


```python
        elif guess.isdigit():
            print("\n!!Invalid input use only words not numbers\n")
        else:
            print("\nIncorrect. The correct word was:- ")
            self.display_in_box(word.capitalize())

    def play_game(self):
        """
        Start and manage the word sorting game.

        Returns:
            None
        """
        print("Welcome to Word Sorting Game!")
        print("Unscramble the words to earn points.")
        print("Enter 'quit' to end the game.\n")
        i = 0
        while True:
            check = self.play_round(i)
            i = 1
            if check == 0:
                print(f"Your final score is:- {self.score}")
                break


# Load words from the JSON file
file = open("cach.json", "r")
words_list = json.load(file)
file.close()
```



```python
            check = self.play_round(i)
            i = 1
            if check == 0:
                print(f"Your final score is:- {self.score}")
                break


# Load words from the JSON file
file = open("cach.json", "r")
words_list = json.load(file)
file.close()

# Create an instance of WordSortingGame and start the game
game_object = WordSortingGame(words_list)
game_object.play_game()
```

Output:-



```
╰ /usr/local/bin/python3 "/Users/shadow/Library/CloudStorage/OneDrive-columbiacollege.ca/Learning/Python/Assignme
/Final.py"
Welcome to Word Sorting Game!
Unscramble the words to earn points.
Enter 'quit' to end the game.

Scrambled word:-
┌───┐ ┌───┐ ┌───┐ ┌───┐ ┌───┐ ┌───┐ ┌───┐
│ N │ │ O │ │ C │ │ U │ │ R │ │ I │ │ N │
└───┘ └───┘ └───┘ └───┘ └───┘ └───┘ └───┘
Enter your guess:
```

User Interface: The system provides a user-friendly interface with the following options:

Quit: To exit the program.

Enter your guess: Sorted word.

Developer Notes: The code is designed to handle various user inputs and provides appropriate error messages for invalid inputs. The program runs continuously until the user chooses to quit.

Conclusion: This concludes the overview of the Word Sorting Game implemented in Python. The game provides a fun and engaging way to test one's vocabulary and spelling skills, with opportunities for further enhancement, such as adding time limits, score multipliers, or hints.

The word sorting game is a simple and entertaining way to challenge oneself and improve vocabulary and spelling skills. With further enhancements, the game can become a valuable tool for language learning and development.