The code provided implements a text-based adventure game set in a secret cavern populated with ghost-type Pokemon. The game begins with an introduction that immerses the player in the cavern's gloomy and ominous environment. The goal of the game is to survive as long as possible, defeat as many opponents as possible, and earn the highest possible score based on the distance travelled through the cave.

The Character class, which depicts a Pokemon character, is at the heart of the game. Each Character object includes various characteristics, including name (the Pokemon's name), hp (hit points, which indicate the Pokemon's health), and attack\_power (the strength of the Pokemon's attacks). Furthermore, the original\_hp attribute stores the Pokemon's original hit points, which is utilised to reset the Pokemon following battles.

The Character class also includes several methods. The is\_alive method checks a Pokemon's current hit points (self.hp) against zero to see if it's alive. A Pokemon can attack another Pokemon using the attack method. It generates a random damage value between 1 and the attack power of the attacking Pokemon, damages the target Pokemon using the receive\_damage method, and saves the battle log using the save\_log function.

The defend method allows a Pokemon to defend itself, lowering the amount of damage it takes. The special\_attack method indicates a strong attack with a 20% probability of instantaneously defeating the target Pokemon while ignoring its remaining hit points. If the special attack fails, the miss is recorded. The receive\_damage method deducts the specified damage from the Pokemon's hit points, preventing them from falling below zero. If the Pokemon loses all of its HP points, it is regarded to have fainted, and the corresponding log is presented.

The reset method restores a Pokemon's hit points to their original value, allowing it to be utilised in battle again. The display\_stats method displays a Pokemon's current hit points.

The choose\_character function asks the player to choose a Pokemon from their inventory. It shows the available Pokemon along with their attributes and allows the player to select one by entering the corresponding number. If an invalid input is given, the function calls itself recursively until a valid choice is made. When a Pokemon is chosen, a confirmation message appears.

The battle function handles the mechanics of the battle between the player's chosen Pokemon and a randomly picked enemy Pokemon. It starts with a log that shows the appearance of the opposing Pokemon. The function enters a loop that lasts as long as both the player's and the enemy's Pokemon remain alive. The current stats of both Pokemon are displayed on each iteration. The player is then given the option of attacking, defending, employing a special attack, or running away.

The corresponding procedures of the player's Pokemon and the enemy Pokemon are executed depending on the action selected. The opponent Pokemon attacks if the player's Pokemon is still alive after their action. The battle will continue until either the player's or the opposing Pokemon faints. When a player's Pokemon passes out, it is withdrawn from their roster. If the player has no more Pokemon, the game will stop and an appropriate log will be displayed. Otherwise, the player is requested to select a different Pokemon from their roster and resume the battle.

The save\_log function adds the specified log message to the file "poke-battle\_log.txt."

The Adventure function implements the main game loop. It sets the global counter variable to zero and creates two lists: Your\_pokemon, which represents the player's Pokemon roster, and Ghosts, which represents the available enemy Pokemon. The function gives the player the option of either