

Mines

Here's how the game works:

1. The user is prompted to enter the number of mines they want, which should be between 1 and 24.
2. The program randomly generates the specified number of mines, each represented by a unique number between 1 and 25, and stores them in a list called mines.
3. The user is then asked if they want to play the game.
4. If the user chooses to play (by entering 'y'), they are prompted to enter a number.
5. The program checks if the user's number is in the mines list:
 - a. If the number is not in the mines list, it is added to the diamonds list (if it hasn't been selected before).
 - b. If the number is in the mines list, the game ends, and the program displays the message "bomb" and "you lose". It also prints the sorted lists of mines and diamonds.
6. If the user chooses not to play (by entering 'n'), the program prints the sorted lists of mines and diamonds and exits.

The game continues until the user either hits a mine or decides not to play anymore.

The program uses lists to keep track of the mines and diamonds, and it employs random number generation to create the mines. The user's input is validated to ensure it falls within the specified range (1 to 24 for the number of mines, and 1 to 25 for the chosen number).

Overall, the mines_game.py file provides a simple implementation of a game where the user tries to find diamonds while avoiding mines, with the mines being randomly generated and the user's choices being tracked and displayed.