

```

import os
import random

def init():
    global animal, wrongLettersList, guessedLettersList, lives, win, animalsList
    lives = 8
    win = False

    # animals.txt by skjorrface (2018).
    https://github.com/skjorrface/animals.txt/blob/master/animals.txt
    # the text file has been modified from the original version
    with open("animals.txt", "r") as file:
        animalsList = [animal.lower().strip()
                       for animal in file.read().split("\n")]
        # open the animals file list and put them in a list in lowercase.

    for name in animalsList:
        if not validName(name):
            animalsList.remove(name)
            # remove any animal names that are not entirely alphabetic characters (+
            # spaces and hyphens)
            # also make sure blank lines in the file are removed, otherwise there will be
            # a case where the "animal" is nothing

    animal = random.choice(animalsList)
    wrongLettersList = []

    # this SELECTED list (below) holds guessed letters, spaces, hyphens, and blanks for
    # the word being guessed
    # this list is the visual representation of the word being guessed and determines the
    # win condition
    # a list is used because it is mutable and easier to manipulate and compare
    guessedLettersList = [
        '_' if letter.isalpha() else letter
        for letter in animal
    ]

    # a valid name is one that contains only alphabetic characters, spaces, and hyphens
    def validName(name):
        return name.replace(" ", "").replace("-", "").isalpha()

```



```

print(
    f"Out of {len(animalsList)} animals, try your best and guess the right one!")
optionInput([
    {
        'label': "Play",
        'function': [gameplay]
    },
    {
        'label': "Guidebook",
        'function': [guidebook]
    },
], menu)

```

```

def guidebook():
    header()
    print("""Here are the basic rules and features of the game:

```

1. Guess one letter at a time. If the letter is correct, it will be revealed in the word.
2. You can also guess the entire word, but it must match the animal's name exactly.
3. Animals' names are not case-sensitive, and only contain alphabetic characters, spaces, and hyphens.
4. You have 8 lives. If you guess incorrectly 9 times, the game is over.
5. The number of lives remaining and the incorrect letters will also be displayed.

```

Now, enjoy the game!""")
optionInput([
    {
        'label': "Back to Menu",
        'function': [menu]
    }
], guidebook)

```

```

def enterToContinue():
    input("\n\nHit Enter to continue.")

```

```

# this SELECTED function (below) checks if the letter is in the word; if true, it
replaces "_" with the letter in the guessed letters list
# if the letter is not in the word, it appends the letter to the wrong letters list (if
not already there)
# the function returns True if the letter is in the word, and False otherwise

```

```

# this function contains iteration, selection, and sequencing
def checkLetter(letter):
    for i in range(len(animal)):
        if animal[i] == letter:
            guessedLettersList[i] = letter
        elif letter not in (animal + "".join(wrongLettersList)):
            wrongLettersList.append(letter)
    return letter in animal

def gameplay():

    def core():
        header()
        if lives >= 0:
            print("Enter 'menu' to go back to menu.\nEnter 'exit' to exit the game.")
            divider()
        print(f"Lives: {lives} {'(last chance!)' if lives == 0 else ''}\n\n"
              ) if lives >= 0 else print("No more lives left...\n\n")
        print(" ".join(guessedLettersList))

    global lives, guessedLettersList, win

    while lives >= 0 and not win:
        core()
        if len(wrongLettersList) > 0:
            print(f"\n\nWrong letters: {' '.join(wrongLettersList)}")
        userInput = input(
            "\n\nGuess a letter (or the whole name): ").lower().strip()

        if not validName(userInput):
            divider()
            print(
                "Please enter a letter or the name of the animal! You are lucky you do
not lose a live for this."
            )
            enterToContinue()

        # assuming the user is entering the whole name of the animal or defined keywords
        elif len(userInput) > 1:
            win = userInput == animal
            if userInput == "menu":
                init()

```

```

        menu()
    if userInput == "exit":
        exit()
    if not win:
        divider()
        print("Unfortunately, that is not the animal you are looking for!")
        enterToContinue()
    else:
        for i in userInput:
            # to have the blanks filled (that is, for visual purpose only)
            checkLetter(i) # <- where the SELECTED function is called
else:
    if checkLetter(userInput): # <- where the SELECTED function is called
        win = "".join(guessedLettersList) == animal
        # check if the name is guessed completely, which would break the while
loop if True
    else:
        lives -= 1
        divider()
        print("Uh oh, the letter is not in the name of the animal!")
        enterToContinue()

core()
print("\n\n" +
      ("Congratulations! You've guessed the correct animal!" if win else
       f"Too bad! The animal you were supposed to guess was '{animal}'"))
optionInput([
    {
        'label': "Restart",
        'function': [init, gameplay]
    },
    {
        'label': "Back to Menu",
        'function': [init, menu]
    },
], gameplay)

if __name__ == "__main__":
    init()
    menu()

```