Project Specification

Project Name: “Game Collection”

Description:

This project, named "Game Collection", is a collection of fun, engaging games developed in Python. It is designed for users to select and play games from a list of choices. This document aims to provide a comprehensive understanding of the project's structure, the games included, the code involved, and various Python functionalities used in building the project.

The default games included are:

1. Mastermind: A classic game where the player tries to guess a random 4-digit number. Feedback is given on each guess to assist the player in making subsequent guesses.
2. Jumbled Word Game: Players are given a jumbled word, and they must unscramble it.
3. Word Guessing Game: Like 'Hangman,' players guess characters in this game for a hidden word.

How to use?

After successfully installing the project, run the following command in your terminal or command prompt:

```
python main.py
```

Upon launching the program, you'll be presented with a main menu showing a list of games available:

1. Mastermind
2. Jumbled Word Game
3. Word Guessing Game
4. Quit

To select a game, simply input the corresponding number.

Mastermind:

Objective: Guess the correct sequence of colors.

1. You have a certain number of attempts to guess the correct sequence.
2. After each guess, you'll receive feedback in the form of colored pegs

Jumbled Word Game:

Objective: Rearrange the scrambled letters to form a valid word.

1. A jumbled word will be shown on the screen.
2. Input your guess and check if it's correct.

Word Guessing Game:

Objective: Guess the word with a limited number of attempts.

1. You'll be shown a series of underscores representing each letter of the word.
2. With each guess, correctly guessed letters will be revealed.

Exiting a Specific Game:

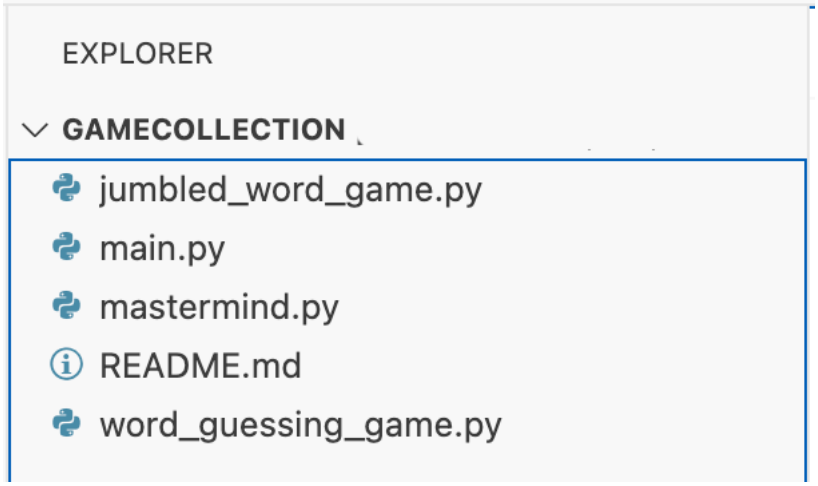
For most games, you can press '**Q**' or '**q**' anytime to quit and return to the main menu.

If a game ends (win or lose), you'll be presented with an option to play again or return to the main menu.

Quitting the Game:

To quit the entire Game Collection, select "**4. Quit**" from the main menu or simply close the terminal or command prompt.

Folder Structure:



Prebuilt Libraries/Packages used:

1. **random:**

   This module implements pseudo-random number generators for various distributions. Functions like randint and choice are used from this module.

   a. randint(a, b): Return a random integer between a and b (inclusive).
   b. choice(sequence): Return a random element from the sequence.
   c. More information about this library can be found on https://pypi.org/project/random2/

2. **subprocess:**

   Allows spawning new processes, connecting to their input/output/error pipes, and obtaining their return codes.

   a. run(args): Used to run the shell commands. In this project, it is used to execute Python game files.
   b. More information about this library can be found on https://docs.python.org/3/library/subprocess.html and https://pypi.org/project/subprocess.run/

Summary

This Game Collection project is a delightful ensemble of various games designed for pure entertainment. Written in Python, it leverages various libraries and constructs. The code is modular, with each game having its Python file, making it easier to understand and manage.

File 1: main.py

Code:

```
import subprocess


def main():
    # Start an infinite loop to keep displaying the menu until the user decides to quit.
    while True:
        # Printing the options for the user to select the game.
        print("Choose a game to play:")
        print("1. Mastermind")
        print("2. Jumbled Word Game")
        print("3. Word Guessing Game")
        # print("4. 2048 Game")
        print("4. Quit")

        # Taking the user's choice.
        choice = input("Enter your choice (1/2/3/4/5): ")

        # Based on the user's choice, run the corresponding game using subprocess.
        # subprocess.run() is used to run shell commands, in this case, to execute the Python
game files.

        if choice == "1":
            # Run the mastermind game.
            subprocess.run(["python", "mastermind.py"])
        elif choice == "2":
            # Run the jumbled word game.
            subprocess.run(["python", "jumbled_word_game.py"])
        elif choice == "3":
            # Run the word guessing game.
            subprocess.run(["python", "word_guessing_game.py"])
        # elif choice == "4":
        #     # Run the 2048 game.
        #     subprocess.run(["python", "2048.py"])
        elif choice == "4":
            # Exit the loop, thereby ending the program.
            print("Thanks for playing!")
            break
        else:
            # If the user enters a choice not listed in the menu, inform them and display the
menu again.
            print("Invalid choice. Please choose again.")


# The code within this conditional block will only execute if main.py is run directly (and not
imported elsewhere).
if __name__ == "__main__":
    main()
```

File 2: mastermind.py

Code:

```
import random

# the .randrange() function generates
# a random number within the specified range.
num = random.randrange(1000, 10000)

n = int(input("Guess the 4 digit number:"))

# condition to test equality of the
# guess made. Program terminates if true.
if(n == num):
    print("Great! You guessed the number in just 1 try! You're a Mastermind!")
else:
    # ctr variable initialized. It will keep count of
    # the number of tries the Player takes to guess the number.
    ctr = 0

    # while loop repeats as long as the Player
    # fails to guess the number correctly.
    while(n != num):
        # variable increments every time the loop
        # is executed, giving an idea of how many
        # guesses were made.
        ctr += 1

        count = 0

        # explicit type conversion of an integer to
        # a string in order to ease extraction of digits
        n = str(n)

        # explicit type conversion of a string to an integer
        num = str(num)

        # correct[] list stores digits which are correct
        correct = []

        # for loop runs 4 times since the number has 4 digits.
        for i in range(0, 4):
            # checking for equality of digits
            if(n[i] == num[i]):
                # number of digits guessed correctly increments
                count += 1
                # hence, the digit is stored in correct[].
                correct.append(n[i])
            else:
                continue

        # when not all the digits are guessed correctly.
        if (count < 4) and (count != 0):
            print("Not quite the number. But you did get ",
```

```
            count, " digit(s) correct!")
        print("Also these numbers in your input were correct.")

        for k in correct:
            print(k, end=' ')

        print('\n')
        print('\n')
        n = int(input("Enter your next choice of numbers: "))

    # when none of the digits are guessed correctly.
    elif(count == 0):
        print("None of the numbers in your input match.")
        n = int(input("Enter your next choice of numbers: "))

if n == num:
    print("You've become a Mastermind!")
    print("It took you only", ctr, "tries.")
```

File 3: word_guessing_game.py

Code:

```
import random
# library that we use in order to choose
# on random words from a list of words

name = input("What is your name? ")

# Here the user is asked to enter the name first

print("Good Luck ! ", name)

words = ['rainbow', 'computer', 'science', 'programming',
        'python', 'mathematics', 'player', 'condition',
        'reverse', 'water', 'board', 'geeks']

# Function will choose one random
# word from this list of words
word = random.choice(words)


print("Guess the characters")

guesses = ''

# any number of turns can be used here
turns = 12


while turns > 0:

    # counts the number of times a user fails
    failed = 0

    # all characters from the input
    # word taking one at a time.
    for char in word:

        # comparing that character with
        # the character in guesses
        if char in guesses:
            print(char, end=" ")

        else:
            print("_")

            # for every failure 1 will be
            # incremented in failure
            failed += 1

    if failed == 0:
        # user will win the game if failure is 0
        # and 'You Win' will be given as output
```

```
    print("You Win")

    # this print the correct word
    print("The word is: ", word)
    break

# if user has input the wrong alphabet then
# it will ask user to enter another alphabet
print()
guess = input("guess a character:")

# every input character will be stored in guesses
guesses += guess

# check input with the character in word
if guess not in word:

    turns -= 1

    # if the character doesn't match the word
    # then "Wrong" will be given as output
    print("Wrong")

    # this will print the number of
    # turns left for the user
    print("You have", + turns, 'more guesses')

    if turns == 0:
        print("You Loose")
```

File 4: jumbled_word_game.py

Code:

```
# Python program for jumbled words game.

# import random module
import random


# function for choosing random word.
def choose():
    # list of word
    words = ['rainbow', 'computer', 'science', 'programming',
             'mathematics', 'player', 'condition', 'reverse',
             'water', 'board', 'geeks']

    # choice() method randomly choose
    # any word from the list.
    pick = random.choice(words)

    return pick


# Function for shuffling the
# characters of the chosen word.
def jumble(word):
    # sample() method shuffling the characters of the word
    random_word = random.sample(word, len(word))

    # join() method join the elements
    # of the iterator(e.g. list) with particular character .
    jumbled = ''.join(random_word)
    return jumbled


# Function for showing final score.
def thank(p1n, p2n, p1, p2):
    print(p1n, 'Your score is :', p1)
    print(p2n, 'Your score is :', p2)

    # check_win() function calling
    check_win(p1n, p2n, p1, p2)

    print('Thanks for playing...')


# Function for declaring winner
def check_win(player1, player2, p1score, p2score):
    if p1score > p2score:
        print("winner is :", player1)
    elif p2score > p1score:
        print("winner is :", player2)
    else:
        print("Draw..Well Played guys..")
```

```
# Function for playing the game.
def play():
    # enter player1 and player2 name
    p1name = input("player 1, Please enter your name :")
    p2name = input("Player 2 , Please enter your name: ")

    # variable for counting score.
    pp1 = 0
    pp2 = 0

    # variable for counting turn
    turn = 0

    # keep looping
    while True:

        # choose() function calling
        picked_word = choose()

        # jumble() function calling
        qn = jumble(picked_word)
        print("jumbled word is :", qn)

        # checking turn is odd or even
        if turn % 2 == 0:

            # if turn no. is even
            # player1 turn
            print(p1name, 'Your Turn.')

            ans = input("what is in your mind? ")

            # checking ans is equal to picked_word or not
            if ans == picked_word:

                # incremented by 1
                pp1 += 1

                print('Your score is :', pp1)
                turn += 1

            else:
                print("Better luck next time ..")

                # player 2 turn
                print(p2name, 'Your turn.')

                ans = input('what is in your mind? ')

                if ans == picked_word:
                    pp2 += 1
                    print("Your Score is :", pp2)
```

```
            else:
                print("Better luck next time...correct word is :", picked_word)

                c = int(input("press 1 to continue and 0 to quit :"))

                # checking the c is equal to 0 or not
                # if c is equal to 0 then break out
                # of the while loop o/w keep looping.
                if c == 0:
                    # thank() function calling
                    thank(p1name, p2name, pp1, pp2)
                    break
        else:
            # if turn no. is odd
            # player2 turn
            print(p2name, 'Your turn.')
            ans = input('what is in your mind? ')

            if ans == picked_word:
                pp2 += 1
                print("Your Score is :", pp2)
                turn += 1
            else:
                print("Better luck next time.. :")
                print(p1name, 'Your turn.')
                ans = input('what is in your mind? ')

                if ans == picked_word:
                    pp1 += 1
                    print("Your Score is :", pp1)
                else:
                    print("Better luck next time...correct word is :", picked_word)

                    c = int(input("press 1 to continue and 0 to quit :"))

                    if c == 0:
                        # thank() function calling
                        thank(p1name, p2name, pp1, pp2)
                        break

            c = int(input("press 1 to continue and 0 to quit :"))
            if c == 0:
                # thank() function calling
                thank(p1name, p2name, pp1, pp2)
                break


# Driver code
if __name__ == '__main__':

    # play() function calling
    play()
```

Output for the Program & Playing the Game:

PASTE THE SCREEN SHOTS OF THE PROJECT

BIBILIOGRAPHY