# **PF final project**

# **report**

**Main code of the actual game:**

import hangmanimages

import random

import time

def chooseWord():

# inFile: file

inFile = open("hangmanwords.txt", 'r')

# line: string

line = inFile.readline()

# wordlist: list of strings

wordlist = line.split()

global word

word=random.choice(wordlist)

#print(word)

chooseWord()

# The parameters we require to execute the game:

def main():

#declaring global variables so they can be used outside this function as well.

global count

global display

global word

global already\_guessed

global length

global play\_game

global limit

length = len(word)

count = 0

display = '\_' \* length

already\_guessed = []

play\_game = ""

def func():

global word

if word == '\_' \* length:

if user\_setting==1:

print("score = 200 points")

elif user\_setting==2:

print("score = 400 points")

elif user\_setting==3:

print("score = 600 points")

elif user\_setting==4:

print("score = 800 points")

elif user\_setting==5:

print("score = 1000 points")

# A loop to re-execute the game when the first round ends:

def play\_loop():

global play\_game

play\_game = input("Would you like to play again? y = yes, n = no \n")

while play\_game not in ["y", "n","Y","N"]:

play\_game = input("Would you like to play again? y = yes, n = no \n")

if play\_game == "y":

main()

elif play\_game == "n":

print("Thank you for playing! We hope that youll come back soon!")

guessed\_display\_tup=()

guessed\_display=set(guessed\_display\_tup)

# Initializing all the conditions required for the game:

def hangman():

#global variables need to be declared in each function we are using them:

global count

global display

global word

global already\_guessed

global play\_game

limit = number\_of\_lives

guess = input("This is the Hangman Word: " + display + " Enter your guess: \n")

#using guess.strip() to remove any spaces in the string

guessed\_display.add(guess)

guess = guess.strip()

if len(guess.strip()) == 0 or len(guess.strip()) >= 2:

print("Invalid Input, Try a letter\n")

hangman()

elif guess in word:

already\_guessed.extend([guess])

index = word.find(guess)

word = word[:index] + "\_" + word[index + 1:]

display = display[:index] + guess + display[index + 1:]

print(display + "\n")

print("already guessed: " + str(guessed\_display))

elif guess in already\_guessed:

print("Try another letter.\n")

else:

count += 1

if limit==10:

if guess not in word:

if count == 1:

time.sleep(1)

print(hangmanimages.dict["first"])

print("Wrong guess. " + str(limit - count) + " guesses remaining\n")

elif count == 2:

time.sleep(1)

print(hangmanimages.dict["second"])

print("Wrong guess. " + str(limit - count) + " guesses remaining\n")

elif count == 3:

time.sleep(1)

print(hangmanimages.dict["third"])

print("Wrong guess. " + str(limit - count) + " guesses remaining\n")

elif count == 4:

time.sleep(1)

print(hangmanimages.dict["fourth"])

print("Wrong guess. " + str(limit - count) + " guesses remaining\n")

elif count == 5:

time.sleep(1)

print(hangmanimages.dict["fifth"])

print("Wrong guess. " + str(limit - count) + " guesses remaining\n")

elif count == 6:

time.sleep(1)

print(hangmanimages.dict["sixth"])

print("Wrong guess. " + str(limit - count) + " guesses remaining\n")

elif count == 7:

time.sleep(1)

print(hangmanimages.dict["seventh"])

print("Wrong guess. " + str(limit - count) + " guesses remaining\n")

elif count == 8:

time.sleep(1)

print(hangmanimages.dict["eighth"])

print("Wrong guess. " + str(limit - count) + " guesses remaining\n")

elif count == 9:

time.sleep(1)

print(hangmanimages.dict["ninth"])

print("Wrong guess. " + str(limit - count) + " guesses remaining\n")

elif count == 10:

time.sleep(1)

print(hangmanimages.dict["tenth"])

print("Wrong guess. " + str(limit - count) + " guesses remaining\n")

print("The word was:",already\_guessed,word)

play\_loop()

elif limit==8:

if guess not in word:

if count == 1:

time.sleep(1)

print(hangmanimages.dict["first"])

print("Wrong guess. " + str(limit - count) + " guesses remaining\n")

elif count == 2:

time.sleep(1)

print(hangmanimages.dict["second"])

print("Wrong guess. " + str(limit - count) + " guesses remaining\n")

elif count == 3:

time.sleep(1)

print(hangmanimages.dict["fourth"])

print("Wrong guess. " + str(limit - count) + " guesses remaining\n")

elif count == 4:

time.sleep(1)

print(hangmanimages.dict["fifth"])

elif count == 5:

time.sleep(1)

print(hangmanimages.dict["sixth"])

print("Wrong guess. " + str(limit - count) + " guesses remaining\n")

elif count == 6:

time.sleep(1)

print(hangmanimages.dict["eighth"])

print("Wrong guess. " + str(limit - count) + " guesses remaining\n")

elif count == 7:

time.sleep(1)

print(hangmanimages.dict["ninth"])

print("Wrong guess. " + str(limit - count) + " guesses remaining\n")

elif count == 8:

time.sleep(1)

print(hangmanimages.dict["tenth"])

print("Wrong guess. " + str(limit - count) + " guesses remaining\n")

print("The word was:",already\_guessed,word)

play\_loop()

elif limit==6:

if guess not in word:

if count == 1:

time.sleep(1)

print(hangmanimages.dict["second"])

print("Wrong guess. " + str(limit - count) + " guesses remaining\n")

elif count == 2:

time.sleep(1)

print(hangmanimages.dict["fourth"])

print("Wrong guess. " + str(limit - count) + " guesses remaining\n")

elif count == 3:

time.sleep(1)

print(hangmanimages.dict["fifth"])

elif count == 4:

time.sleep(1)

print(hangmanimages.dict["seventh"])

print("Wrong guess. " + str(limit - count) + " guesses remaining\n")

elif count == 5:

time.sleep(1)

print(hangmanimages.dict["ninth"])

print("Wrong guess. " + str(limit - count) + " guesses remaining\n")

elif count == 6:

time.sleep(1)

print(hangmanimages.dict["tenth"])

print("Wrong guess. " + str(limit - count) + " guesses remaining\n")

print("The word was:",already\_guessed,word)

play\_loop()

elif limit==4:

if guess not in word:

if count == 1:

time.sleep(1)

print(hangmanimages.dict["third"])

print("Wrong guess. " + str(limit - count) + " guesses remaining\n")

elif count == 2:

time.sleep(1)

print(hangmanimages.dict["sixth"])

print("Wrong guess. " + str(limit - count) + " guesses remaining\n")

elif count == 3:

time.sleep(1)

print(hangmanimages.dict["eighth"])

print("Wrong guess. " + str(limit - count) + " last guess remaining\n")

elif count == 4:

time.sleep(1)

print(hangmanimages.dict["tenth"])

print("Wrong guess. You are hanged!!!\n")

print("The word was:",already\_guessed,word)

play\_loop()

elif limit==2:

if guess not in word:

if count == 1:

time.sleep(1)

print(hangmanimages.dict["fifth"])

print("Wrong guess. " + str(limit - count) + " guesses remaining\n")

elif count == 2:

time.sleep(1)

print(hangmanimages.dict["tenth"])

print("Wrong guess. You are hanged!!!\n")

print("The word was:",already\_guessed,word)

play\_loop()

if word == '\_' \* length:

print("Congrats! You have guessed the word correctly!")

func()

play\_loop()

elif count != limit:

hangman()

#Welcoming the user to the game and using time.sleep() to increase interactivity:

print("\nWelcome to our Hangman game \n")

time.sleep(2)

user\_choice=input("would you like to play? press (1) if yes or (0) if no")

if user\_choice=="1":

print("We hope that you have fun playing this :)")

time.sleep(2)

name=input("Enter your name: ")

print("Hello " + name + "! Best of Luck!")

print("Loading...")

time.sleep(3)

print("The game is about to start!\n Let's play Hangman!")

time.sleep(3)

print("There are multiple difficulty settings shown below:")

print("\t1. Beginner (10 lives)")

print("\t2. Intermediate (8 lives)")

print("\t3. Difficult (6 lives)")

print("\t4. Expert (4 lives)")

print("\t5. Pro Insane(2 lives)")

user\_setting = input("At what difficulty would you like to play: ")

if (str(user\_setting) == "1"):

number\_of\_lives = 10

print("\nYou have chosen %s and will receive %d lives." % ("Beginner", number\_of\_lives))

elif(str(user\_setting) == "2"):

number\_of\_lives = 8

print("\nYou have chosen %s and will receive %d lives." % ("Intermediate", number\_of\_lives))

elif (str(user\_setting) == "3"):

number\_of\_lives = 6

print("\nYou have chosen %s and will receive %d lives." % ("Expert", number\_of\_lives))

elif (str(user\_setting) == "4"):

number\_of\_lives = 4

print("\nYou have chosen %s and will receive %d lives." % ("Advanced", number\_of\_lives))

elif (str(user\_setting) == "5"):

number\_of\_lives = 2

print("\nYou have chosen %s and will receive %d lives." % ("Insane", number\_of\_lives))

else:

number\_of\_lives = 10

print("\nYou have made an invalid selection and will receive %d lives by default." % number\_of\_lives)

# Choose a difficulty level

main()

hangman()

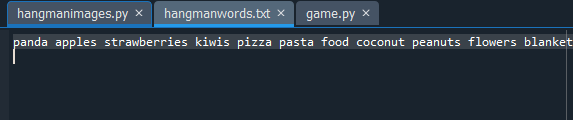
else:

print("Bye bye then...")

print("Hope you'll come play sometime else")

**Word list in external txt file:**

panda apples strawberries kiwis pizza pasta food coconut peanuts flowers blankets cozy yummy great fun imaginary delusional funny quirky rainbows fluffy sparkles pastel hazelnuts almonds pages project passing marker



**dictionary in external file:**

dict ={

"first":"""

\_\_\_\_\_

|

|

|

|

|

|

|

|

\_\_|\_\_ \n""",

"second":"""

\_\_\_\_\_\_

| |

|

|

|

|

|

|

|

\_\_|\_\_""",

"third":"""

\_\_\_\_\_\_

| |

| |

|

|

|

|

|

|

\_\_|\_\_""",

"fourth":"""

\_\_\_\_\_\_

| |

| |

| |

|

|

|

|

|

\_\_|\_\_""",

"fifth":"""

\_\_\_\_\_\_

| |

| |

| |

| |

| O

|

|

|

\_\_|\_\_""",

"sixth":"""

\_\_\_\_\_\_

| |

| |

| |

| |

| O

| |

|

|

\_\_|\_\_""",

"seventh":"""

\_\_\_\_\_\_

| |

| |

| |

| |

| O

| /|

|

|

\_\_|\_\_""",

"eighth":"""

\_\_\_\_\_\_

| |

| |

| |

| |

| O

| /|\

|

|

\_\_|\_\_""",

"ninth":"""

\_\_\_\_\_\_

| |

| |

| |

| |

| O

| /|\

| /

|

\_\_|\_\_""",

"tenth":"""

\_\_\_\_\_\_

| |

| |

| |

| |

| O

| /|\

| / \

|

\_\_|\_\_

"""

}

A computer screen shot of lines and dots

Description automatically generated

**Output:**

If you play and win:

A screenshot of a computer game

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

If you choose not to play:

A screenshot of a computer screen

Description automatically generated

If you play and lose:

A screenshot of a computer program

Description automatically generated

A screenshot of a computer game

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer game

Description automatically generated

A screenshot of a computer game

Description automatically generated

**Explanation of Code:**

1. Importing hangmanimages which stores a dictionary, which has the pattern of the hangman
2. Importing random so that we can randomly choose a word from our txt file. This will allow the game to be different every time it loads
3. Importing the time module so that the game can be more interactive. We use time.sleep throughout so that a break of a certain amount of seconds can be left before the next line is interpreted
4. The next chooseWord function is used to choose a random word using random.choice(). The random word is chosen from the txt file which is opened and read.
5. All words are stored in a single line so, we only do readline once
6. Next a word list is created from the words in the txt file through splitting. This makes it easier for us to choose one word from this list through the random.choice func
7. At the end this function is called so that a random word is saved for us
8. In the next main() function, we declare some random variables:

global count: used to count the users lives as the user guesses wrong

global display: Used to display what letters are left to guess and what position as the user guesses

global word: This is the same word we found from the previous function. In order to use it in this function, we have to call it global again.

global already\_guessed: a empty list which will later contain all the letters weve guessed.

global length: length of word

global play\_game is an empty string which will later be used to input on if the user wants to play again or not

global limit is declared which gives us the limit of the lives the user will get depending on what mode is being played

1. Def func is declared next which would be used to display the users score at the end of the game depending on the mode the user plays in
2. Next the play loop is defined which will be used at the end of the game to input on whether the user wants to replay the game or not
3. Next a set is created. This set will show the user what he has already guessed, so that he can keep track of his guesses and it is easier to play.
4. Next the hangman game is declared in a function. The same variables are declared as global again so that they can be reused. This guess is added to the set we created above to be shown to the user to keep track with his guesses.
5. Guess.strip() is used next to remove any spaces found in the string so that a single word is found
6. The next statement: if len(guess.strip()) == 0 or len(guess.strip()) >= 2:

print("Invalid Input, Try a letter\n") ensures that a single letter is entered by the user only. And if it iss invalid hangman will run again to ask the user their guess again

1. If the guess was already in the word: it is added to the already guessed list. Index variable will be used to find the specific index in the word on which the guess is present. The letter in the word will be replaced by a dash next so that at the end, if all guesses were right, full dashes will be there instead of the word eg: hello=\_\_\_\_\_ so the computer knows the user has won. Display on the other hand, will replace the dashes by letters of the word to show the user their progress. Eg \_\_\_\_\_=h\_\_\_\_
2. If the guess was already guessed before: User is asked to input some other letter.
3. If guess was not present in the word, count is incremented.
4. We import images from hangmanimages.py which stores a dictionary and knows what image to print at every turn the user guesses wrong. If all the guesses are guessed wrong, the whole person is printed.
5. At every turn the number of lives left for the user to guess are printed too
6. At the end if the word=\_\_\_\_ as we said before, the user win. The number of dashes printed must be equal to the length of the word, which is a variable we’ve declared and found before
7. Play loop is called next, to ask the user if they want to play again after loosing
8. If count isn’t equal to the limit, the hangman function is called which allows a loop to occur, asking the user to input their guess again. A recursive function was used here.
9. Next we input from the user if they want to play the game or not. If they choose 1, the following happens:
10. The next part is some statements which are printed for the user at the very start of the game to welcome them etc.
11. Time.sleep is used to the time of 1 or 2s for the game to pause for that much time before the next statements are printed. This is done to make the user experience while playing the game nicer.
12. Next we input what mode the user wouldlike to play for. The modes are shown first in bullet list by the help of tabs. \t is tab and it leaves some space in a string
13. %s is used for string %d is used for numbers. This is used for formatting, to make the coding more convenient for us.
14. Next we tell the computer, how many lives are there for each mode(user\_setting)
15. Next the main and hangman functions are called so that the game can start.
16. If the user chooses 0 at the start, only these statements are printed: print("Bye bye then...")

print("Hope you'll come play sometime else")

**What we used when making this game:**

1. Recursive functions
2. File handling
3. Dictionaries
4. Sets
5. Lists
6. tupples
7. If,elif,else statements
8. Nested if statements
9. While loops
10. Global variables
11. Time module
12. Random module
13. \n,\t,%s,%d
14. Strip function
15. Split function
16. Incrementing count