Project progress Report-2

(Submission Date: 30 October 2020)

Hangman Game using Python IDLE

BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE & ENGINEERING

Submitted By:
Harshit jaiswal(17BCS2482)

Name of the Mentor
Aanchal Karma

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Chandigarh University, Gharuan

1. Introduction:

Hangman is a popular word guessing game where the player attempts to build a missing word by guessing one letter at a time. After a certain number of incorrect guesses, the game ends and the player loses. The game also ends if the player correctly identifies all the letters of the missing word. Hangman is a paper and pencil guessing game for two or more players. One player thinks of a word and the other tries to guess it by suggesting the letters.The word to guess is represented by a row of dashes, giving the number of letters. If the guessing player suggests a letter which occurs in the word, the program writes it in all its correct positions. If the suggested letter does not occur in the word, the other player draws one element of the hangman diagram as a tally mark.

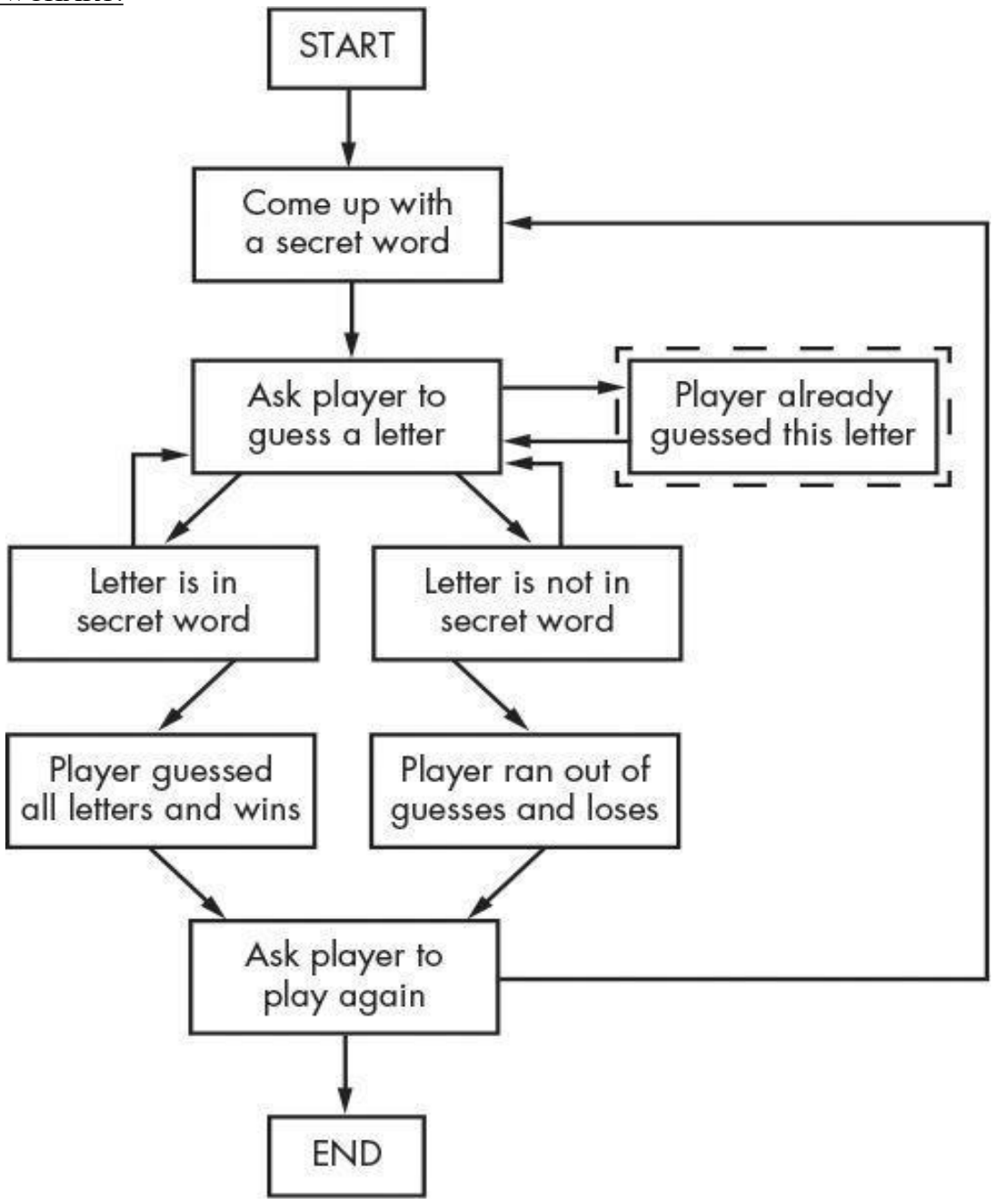
HOW TO PLAY: My code will generate a word which has to be guessed by the player. So,at the output screen will exist marked out blanks (short lines) for each letter of a word.Then the player will guess a letter. If that letter is in the word(s) then the project will write the letter at everyplace it appears, and cross out that letter in the alphabet. If the letter isn't in the word then we cross out the lifelines (which are usually a finite no. of chances) from the list. The player will continue guessing the letters until he can either solve the word (or phrase) or he will end up losing all the lifelines and he will be declared a LOSER.

2. Implementation:

I have decided to use PYTHON language and specifically PyGame toolkit for the GUI package. Initially I was confused between Tkinter and PyGame , as python itself is very new to me. And to install the python pyGame you have to install the pyGame package in your python IDLE, By typing the following command in the command prompt ”py -m pip **install** -U **pygame** --user”. But then,thought of utilizing the classes and modules of this package for my gaming environment. This game will mainly play with the 26 letters of the English language , so my input data will be just letters and as output the player comes to know whether he has won , or else if lost , what was the word which he missed to guess correctly.Thus , there will be a list of words in store with the program from which a player will be asked to guess a word. My estimated lines of code is 200 but it may be little more depending upon how much well I can program.I need 2 weeks time for theproject as I am new to python and specially GUI code .Currently I have only done some homework about what my project will do , exactly what it will accomplish ,but I am yet to start with coding. I hope and will try my best to complete it before the scheduled deadline.

3. Design:

FLOWCHART:



-> So, As the game starts. The game program comes up with a secret word or sentence that is blank and you have to fill in that blank.

-> Then, In the next step the program will ask you to input the word or sentence guessed by you.

-> Now, The program will check whether the word or sentence entered by you is correct or not.

-> If the guess is correct and it matches the secret word then, program will print that player guessed the Correct answer and wins the game. But, if the answer is not correct and you ran out of the guesses then you will lose the game.

-> Now, Ask the player whether he/she want to play again the game. If he/she want to play the generate a new secret word or sentence and continue with the steps.

-> END.

4. Screenshot:

```
import pygame
import random
import os
import sys

pygame.init()
winHeight = 480
winWidth = 700
win=pygame.display.set_mode((winWidth,winHeight))
#-------------------------------------#
# initialize global variables/constants #
#-------------------------------------#
BLACK = (0,0, 0)
WHITE = (255,255,255)
RED = (255,0, 0)
GREEN = (0,255,0)
BLUE = (0,0,255)
LIGHT_BLUE = (102,255,255)

btn_font = pygame.font.SysFont("arial", 20)
guess_font = pygame.font.SysFont("monospace", 24)
lost_font = pygame.font.SysFont('arial', 32)
word = ''
buttons = []
guessed = []
hangmanPics = [pygame.image.load('hangman0.png'), pygame.image.load('hangman1.png'), pygame.image.load('hangman2.png'), pygame.image.load('hangman3.png'), pygame.image

limbs = 0


def redraw_game_window():
    global guessed
    global hangmanPics
    global limbs
    win.fill(GREEN)
    # Buttons
    for i in range(len(buttons)):
        if buttons[i][4]:
            pygame.draw.circle(win, BLACK, (buttons[i][1], buttons[i][2]), buttons[i][3])
            pygame.draw.circle(win, buttons[i][0], (buttons[i][1], buttons[i][2]), buttons[i][3] - 2
                               )
            label = btn_font.render(chr(buttons[i][5]), 1, BLACK)
```

```
            win.blit(label, (buttons[i][1] - (label.get_width() / 2), buttons[i][2] - (label.get_height() / 2)))

    spaced = spacedOut(word, guessed)
    label1 = guess_font.render(spaced, 1, BLACK)
    rect = label1.get_rect()
    length = rect[2]

    win.blit(label1,(winWidth/2 - length/2, 400))

    pic = hangmanPics[limbs]
    win.blit(pic, (winWidth/2 - pic.get_width()/2 + 20, 150))
    pygame.display.update()


def randomWord():
    file = open('words.txt')
    f = file.readlines()
    i = random.randrange(0, len(f) - 1)

    return f[i][:-1]


def hang(guess):
    global word
    if guess.lower() not in word.lower():
        return True
    else:
        return False


def spacedOut(word, guessed=[]):
    spacedWord = ''
    guessedLetters = guessed
    for x in range(len(word)):
        if word[x] != ' ':
            spacedWord += '_ '
            for i in range(len(guessedLetters)):
                if word[x].upper() == guessedLetters[i]:
                    spacedWord = spacedWord[:-2]
                    spacedWord += word[x].upper() + ' '
        elif word[x] == ' ':
            spacedWord += ' '
```

```
    return spacedWord


def buttonHit(x, y):
    for i in range(len(buttons)):
        if x < buttons[i][1] + 20 and x > buttons[i][1] - 20:
            if y < buttons[i][2] + 20 and y > buttons[i][2] - 20:
                return buttons[i][5]
    return None


def end(winner=False):
    global limbs
    lostTxt = 'Lose!! press any key to play again..'
    winTxt = 'WINNER!,press any key to play again..'
    redraw_game_window()
    pygame.time.delay(1000)
    win.fill(GREEN)

    if winner == True:
        label = lost_font.render(winTxt, 1, BLACK)
    else:
        label = lost_font.render(lostTxt, 1, BLACK)

    wordTxt = lost_font.render(word.upper(), 1, BLACK)
    wordWas = lost_font.render('The phrase was: ', 1, BLACK)

    win.blit(wordTxt, (winWidth/2 - wordTxt.get_width()/2, 295))
    win.blit(wordWas, (winWidth/2 - wordWas.get_width()/2, 245))
    win.blit(label, (winWidth / 2 - label.get_width() / 2, 140))
    pygame.display.update()
    again = True
    while again:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
            if event.type == pygame.KEYDOWN:
                again = False
    reset()
```

```
def reset():
    global limbs
    global guessed
    global buttons
    global word
    for i in range(len(buttons)):
        buttons[i][4] = True

    limbs = 0
    guessed = []
    word = randomWord()

#MAINLINE


# Setup buttons
increase = round(winWidth / 13)
for i in range(26):
    if i < 13:
        y = 40
        x = 25 + (increase * i)
    else:
        x = 25 + (increase * (i - 13))
        y = 85
    buttons.append([LIGHT_BLUE, x, y, 20, True, 65 + i])
    # buttons.append([color, x_pos, y_pos, radius, visible, char])

word = randomWord()
inPlay = True
```

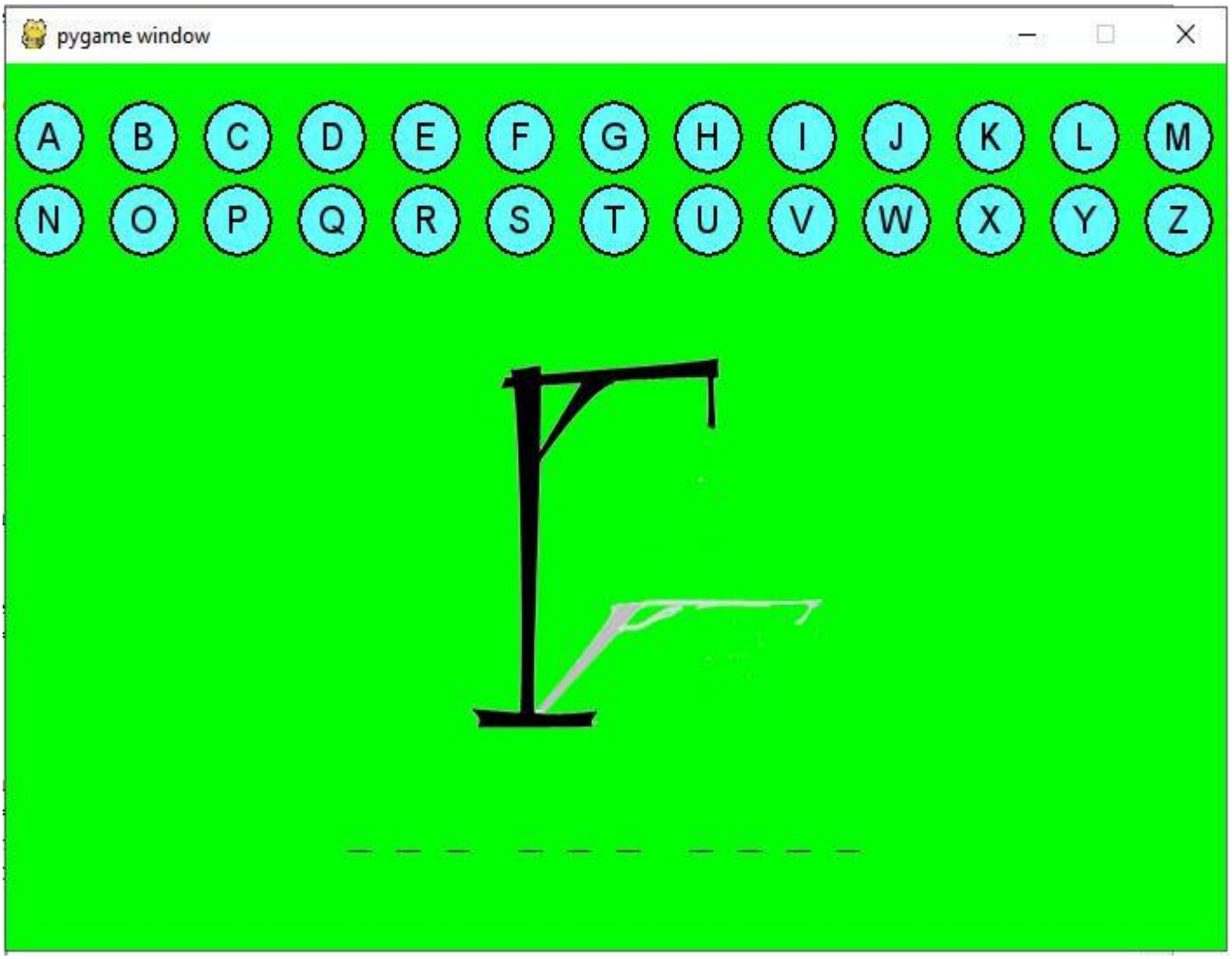
```
while inPlay:
    redraw_game_window()
    pygame.time.delay(10)

    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            inPlay = False
        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_ESCAPE:
                inPlay = False
        if event.type == pygame.MOUSEBUTTONDOWN:
            clickPos = pygame.mouse.get_pos()
            letter = buttonHit(clickPos[0], clickPos[1])
            if letter != None:
                guessed.append(chr(letter))
                buttons[letter - 65][4] = False
                if hang(chr(letter)):
                    if limbs != 5:
                        limbs += 1
                    else:
                        end()
                else:
                    print(spacedOut(word, guessed))
                    if spacedOut(word, guessed).count('_') == 0:
                        end(True)

pygame.quit()
```

Outcomes:

The outcome of the project will be to check your English vocabulary and check your sentence formation in the Hangman game itself



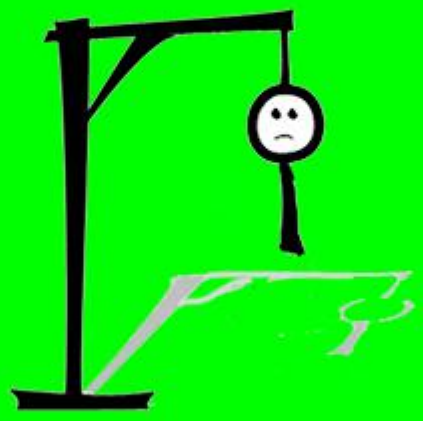
-> So,As you can see in the above picture there will be blanks left for you to fill.

-> In order to fill you need to select a letter from the above given alphabets. And if your guessed letter is incorrect then the hangman will appear and in this manner you will have a maximum of 5 chances left to predict the sentence and if all you 5 prediction of letter is incorrect then the game will end and you lose.

-> You can either close the program or you can play again the game by pressing any key.

B C D E G I J K L M

N O P Q R S U V W X Y Z



H A _ _ _ A _ _ _ _ _ _ _ _

pygame window

Lose!! press any key to play again..

The phrase was:

HANGMAN IS COOL

pygame window

A B C F G H I J K L M

P Q R S T V W X Y Z

O D D O N E O U _

pygame window

WINNER!,press any key to play again..

The phrase was:

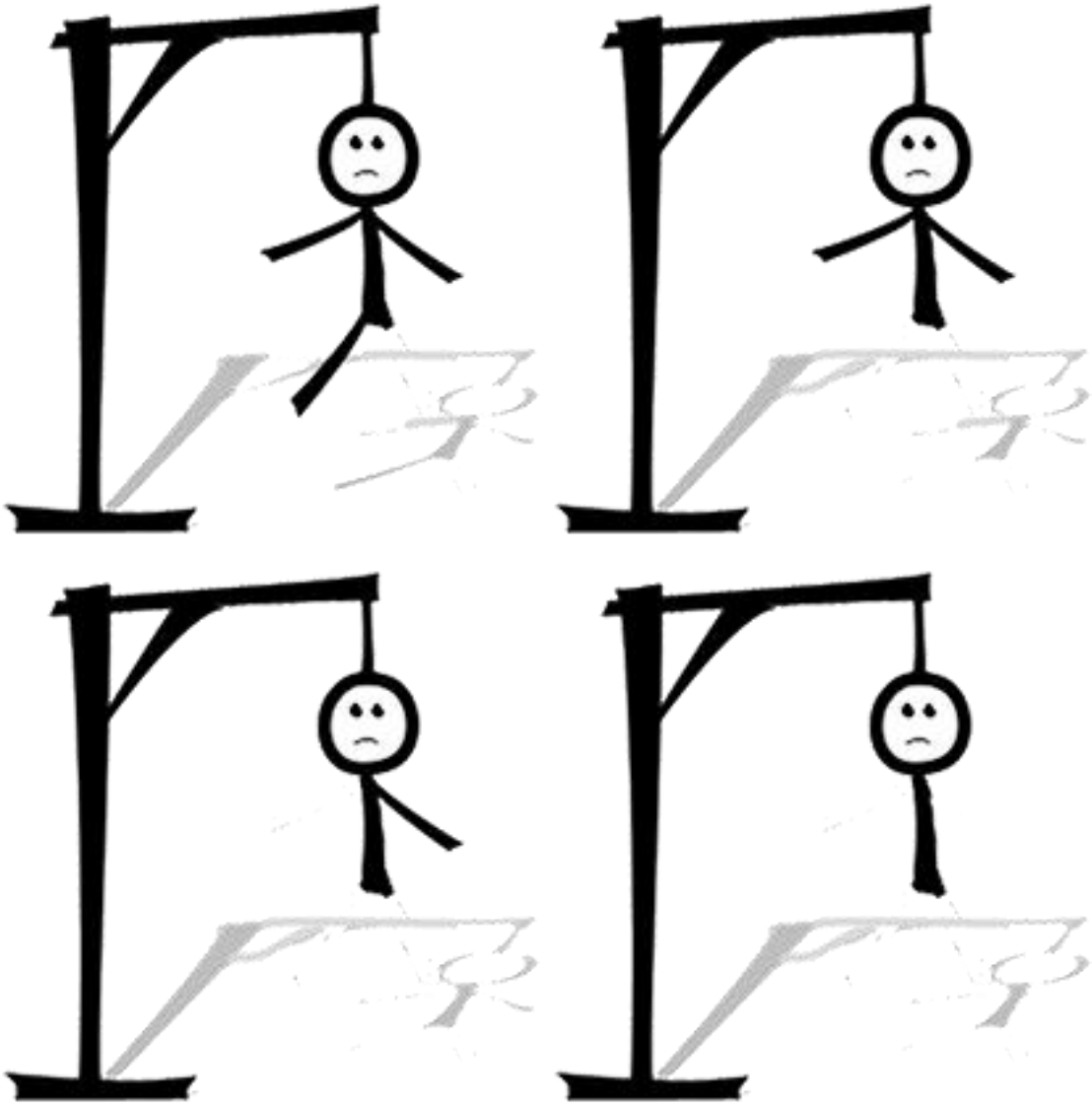
ODD ONE OUT

words - Notepad

File Edit Format View Help

```
the big bang
the pillow feels soft
odd one out
press conference
show me the money
time to go home
that was easy
hangman is cool
zoologist
quadruplets
the sky is blue
```

There is a file named word in which all the phrases are included which are given to the user to guesses. You can add as many phrases as you want.



And for the hangman images you can search for pictures on google and import the pictures on the python program that you are using. Just like I used theses picture in my interactive python hangman game.