

COMPUTER SCIENCE PROJECT

WORDLE WITH PYGAME

W O R D L E

A DAILY WORD GAME

NAME: S.MOKITHA
CLASS: XII
SECTION: A

BONAFIDE CERTIFICATE

Certified to be the Bonafide project work in **COMPUTER SCIENCE**

Done by **S.MOKITHA** , Register number _____

of Class **XII** Section **A** of D.A.V GIRLS SR. SEC. SCHOOL,
GOPALAPURAM, CHENNAI-600 086 during the year 2022 to 2023.

Signature of Principal

Signature of Subject Teacher

School Seal

Submitted for the practical examination held on _____ at D.A.V.
GIRLS SENIOR SECONDARY SCHOOL, GOPALAPURAM, CHENNAI-600
086

External Examiner

Internal Examiner

ACKNOWLEDGEMENT

I would like to offer my sincere thanks to our Principal Mrs. Radha Venkatesh for giving me the opportunity to do this project, which was a wonderful knowledge gaining experience. I also extend my gratitude to our Computer Science teacher Mrs. Uma Thiagarajan for guiding me in the completion of this project successfully. I also thank our lab attender Mrs.Lakshmi in helping me with all the requirements.

INDEX

S.NO	TOPIC	PAGE NO
01.	Aim	1
02.	Files used	2
03.	User defined functions used	3
04.	Program Listing	7
05.	Sample Outputs	25
06.	Bibliography	28

AIM

To create wordle, a word-based game with pygame library which consists of computer graphics and other libraries designed to be used along with python programming language. The game will consist of five windows namely,

- (1) **rules**, where the instructions of the game are displayed
- (2) **menu**, which allows the user to play a new game and choose the complexity level
- (3) **options**, that lets the user select the difficulty level
- (4) **game board** in which the guesses can be entered
- (5) **result**, that displays the game status.

This game will use the display, mouse, draw, font, image and event module of the pygame library and also the sleep function of the time module.

FILES USED

TEXT FILES

- **Easy.txt**

It contains all the words belonging to the easy difficulty level which will be imported when the user selects the said level. The word for that round of the game will be randomly chosen from these imported words.

- **Medium.txt**

It contains all the words belonging to the moderate complexity level which will be imported when the user selects the said level. The word for that round of the game will be randomly chosen from these imported words.

- **Hard.txt**

It contains all the words belonging to the hard difficulty level which will be imported when the user selects the said level. The word for that round of the game will be randomly chosen from these imported words.

- **Rules.txt**

It contains the rules of the game play of wordle which is imported within the rules() function and displayed in the first layer of the game.

USER-DEFINED FUNCTIONS

- **rules()**

It is a void function which initializes a pygame window of 386 x 700 pixels along with the caption 'Wordle' and the icon. It loads, creates the surface object and blits the background sprite, continue button and the text from the rules.txt file imported within the function. Finally, it updates the screen for display.

- **click_rules()**

It is a void function which gets the rectangular area of the continue button image and positions it. If the mouse pointer hovers above that area, the image inflates and when it is clicked, the menu is displayed. Else, the rules() function is called to display the rules window.

- **menu()**

It is a void function which creates a new layer menu. It loads, creates the surface object and blits the background, new game, options and exit button sprites. This function also renders and pastes the title 'WORDLE' on the window. Finally, it updates the screen for display.

- **click_newgame()**

It is a void function which gets the rectangular area of the new game, options and exit button sprites and positions it. If the mouse hovers above any one of the areas, the image inflates. The new game button can be clicked only when the complexity level is chosen and when it is done, the game board and indicators

will be displayed. If the options button is clicked, the `word_click()` function is called and the options window is displayed. If the exit button is clicked, the pygame window is quit. Else, the menu is displayed.

- **`words()`**

It is a void function which is created when the options button of the menu layer is clicked. It loads, creates the surface object and blits the background, easy, medium and hard button sprites.

- **`word_click()`**

It is a void function which gets the rectangular area of the easy, medium and hard button sprites and positions it. If the mouse pointer hovers above any of the areas, the image inflates and when it is clicked, the text file corresponding to that difficulty level is imported and the word for the game is randomly chosen from the contents of the text file. The `menu()` function is called within it and the display goes back to the menu layer.

- **`game_board()`**

It loads, creates the surface object and blits the background sprite consisting of the tiles for the game. This is the third layer of the game.

- **`add_letter(key, x_space, y_space, color_l, color_bg, size)`**

It takes the alphabet, its size, x-coordinate, y-coordinate of the position of the letter, background colour and the colour of the letter as its parameters and

displays the letter within the desired box according to its coordinates. Finally, it updates the screen for display.

- **colour(word, L)**

It takes the correct word and a list consisting of letters entered in one guess as parameters. If the letter is present in the word at the correct position, the tile turns green. If the letter is at the wrong position, it turns yellow. If the letter is not present in the word, the tile turns grey.

- **indicator()**

It is a void function that draws rectangles at the bottom of the screen which resemble indicators and inserts letters inside them using the `add_letter()` function.

- **indicator_color(S, in_x, in_y, l_x, l_y)**

It takes a string of letters of each row of the indicators, x and y coordinates of the position of letters and its background rectangles as parameters. If the letter is present in the word at the correct position, the indicator turns green. If the letter is at the wrong position, it turns yellow. If the letter is not present in the word, the indicator turns grey.

- **result()**

If the guess is right or the player has no more guesses left, it loads, creates

the surface object and blits the background sprite along with some text that shows if the player has won or lost. It also adds two buttons to 'play again' and 'exit'. It returns 'w' if the player has won, 'l' if the player has lost and an empty string otherwise.

- **reset_button()**

It is a void function which gets the rectangular area of the play again and exit button sprites and positions it. If the mouse pointer hovers above any of the areas, the image inflates. When the play again button is clicked, the menu is displayed. If the exit button is clicked, the pygame window is quit

- **delete(x_space, y_space)**

It draws a white solid rectangle at the given coordinates.

- **quit()**

It exits out of the pygame window.

PROGRAM LISTING

```
import pygame,sys,random,time
pygame.init()

def words():
    global easy, medium, hard
    bg=pygame.image.load('menu.png').convert()
    screen.blit(bg,(0,0))
    font=pygame.font.SysFont('timesnewroman',40)
    text=font.render('Difficulty Levels',True,BLACK)
    screen.blit(text,(55,60))
    easy=pygame.image.load('easy.jpg').convert()
    screen.blit(easy,(120,150))
    medium=pygame.image.load('medium.jpg').convert()
    screen.blit(medium,(120,280))
    hard=pygame.image.load('hard.jpg').convert()
    screen.blit(hard,(120,410))
    pygame.display.update()

def word_click():
    global easy, medium, hard, word, new
    rect1=easy.get_rect(topleft=(120,150))
    rect2=medium.get_rect(topleft=(120,280))
    rect3=hard.get_rect(topleft=(120,410))
    while True:
        quit()
```

```

if pygame.mouse.get_pos()[0] in range(rect1.left,rect1.right) and\
pygame.mouse.get_pos()[1] in range(rect1.top,rect1.bottom):
    easy=pygame.transform.scale(easy,(145,107))
    screen.blit(easy,(110,140))
    pygame.display.update()
    if pygame.mouse.get_pressed()[0]==1:
        with open('easy.txt','r') as F:
            L=F.readlines()
            word=L[random.randint(0,len(L)-1)][:5].upper()
        menu()
        time.sleep(1)
        new=True
        return
    elif pygame.mouse.get_pos()[0] in range(rect2.left,rect2.right) and \
pygame.mouse.get_pos()[1] in range(rect2.top,rect2.bottom):
        medium=pygame.transform.scale(medium,(145,107))
        screen.blit(medium,(110,270))
        pygame.display.update()
        if pygame.mouse.get_pressed()[0]==1:
            with open('medium.txt','r') as F:
                L=F.readlines()
                word=L[random.randint(0,len(L)-1)][:5].upper()
            menu()
            time.sleep(1)
            new=True
            return
    elif pygame.mouse.get_pos()[0] in range(rect3.left,rect3.right) and\
pygame.mouse.get_pos()[1] in range(rect3.top,rect3.bottom):
        hard=pygame.transform.scale(hard,(145,107))

```

```

    screen.blit(hard,(110,400))
    pygame.display.update()
    if pygame.mouse.get_pressed()[0]==1:
        with open('hard.txt','r') as F:
            L=F.readlines()
            word=L[random.randint(0,len(L)-1)][:5].upper()
            menu()
            time.sleep(1)
            new=True
            return
    else:
        words()

def add_letter(key,x_space,y_space,color_l,color_bg,size):
    global BLACK,WHITE,LIGHTGRAY
    font=pygame.font.Font('freesansbold.ttf',size)
    text=font.render(key,True,color_l)
    rect=text.get_rect()
    pygame.draw.rect(text,color_bg,rect,1)
    screen.blit(text,(x_space,y_space))
    pygame.display.update()

def colour(word,L):
    global x,y,xb,yb,WHITE,GREEN,GRAY,RED,YELLOW,green
    for k in range(len(L)):
        if L[k] in word:
            if L[k].lower()==word[k].lower():
                pygame.draw.rect(screen,GREEN,pygame.Rect(xb,yb,51,51))
                add_letter(L[k],x,y,WHITE,GREEN,32)

```

```
x+=60
xb+=60
green+=1
D[L[k]]='green'
D1[L[k]]-=1
elif D1[L[k]]>0:
    pygame.draw.rect(screen,YELLOW,pygame.Rect(xb,yb,51,51))
    add_letter(L[k],x,y,WHITE,YELLOW,32)
    x+=60
    xb+=60
    if L[k] not in D:D[L[k]]='yellow'
    D1[L[k]]-=1
else:
    pygame.draw.rect(screen,GRAY,pygame.Rect(xb,yb,51,51))
    add_letter(L[k],x,y,WHITE,GRAY,32)
    x+=60
    xb+=60
    if L[k] not in D:D[L[k]]='gray'
    D1[L[k]]-=1
else:
    pygame.draw.rect(screen,GRAY,pygame.Rect(xb,yb,51,51))
    add_letter(L[k],x,y,WHITE,GRAY,32)
    x+=60
    xb+=60
    if L[k] not in D:D[L[k]]='gray'
y+=58
yb+=60
x,xb=60,47
pygame.display.update()
```

```
def delete(x_space,y_space):
    pygame.draw.rect(screen,WHITE,pygame.Rect(x_space,y_space,30,30))
    pygame.display.update()

def result():
    global reset1,reset2,tc, word
    if green==5:
        screen=pygame.display.set_mode((386,700))
        screen.fill(WHITE)
        confettiup=pygame.image.load('confettiup.jpg').convert()
        screen.blit(confettiup,(0,0))
        confettidown=pygame.image.load('confettidown.jpg').convert()
        screen.blit(confettidown,(0,450))
        font=pygame.font.SysFont('freestylescript',60)
        text=font.render('Your word was right!!',True,GOLD)
        screen.blit(text,(10,170))
        font=pygame.font.SysFont('freestylescript',48)
        text=font.render('The correct word was',True,GOLD)
        screen.blit(text,(10,220))
        text=font.render(word.lower(),True,GOLD)
        screen.blit(text,(293,220))
        font=pygame.font.SysFont('timesnewroman',40)
        text=font.render('Winning Streak:',True,BLACK)
        screen.blit(text,(40,270))
        font=pygame.font.SysFont('timesnewroman',40)
        text=font.render(str(ws),True,BLACK)
        screen.blit(text,(310,270))
        reset1=pygame.image.load('reset1.png').convert()
```

```
screen.blit(reset1,(95,340))
font=pygame.font.SysFont('timesnewroman',30)
text=font.render('Play Again',True,BLACK)
screen.blit(text,(128,355))
reset2=pygame.image.load('reset2.png').convert()
screen.blit(reset2,(95,440))
text=font.render('Exit',True,BLACK)
screen.blit(text,(170,450))
gr='w'
pygame.display.update()
elif tc==6:
    screen=pygame.display.set_mode((386,700))
    screen.fill(WHITE)
    confettiuplost=pygame.image.load('confettiuplost.jpg').convert()
    screen.blit(confettiuplost,(0,0))
    confettidown=pygame.image.load('confettidownlost.jpg').convert()
    screen.blit(confettidown,(0,450))
    font=pygame.font.SysFont('freestylescript',48)
    text=font.render('Oops!! Your guess was wrong!!',True,RED)
    screen.blit(text,(10,170))
    text=font.render('The correct word was',True,RED)
    screen.blit(text,(10,220))
    text=font.render(word.lower(),True,RED)
    screen.blit(text,(293,220))
    font=pygame.font.SysFont('timesnewroman',40)
    text=font.render('Winning Streak:',True,BLACK)
    screen.blit(text,(40,270))
    font=pygame.font.SysFont('timesnewroman',40)
    text=font.render(str(ws),True,BLACK)
```



```

    screen.blit(text,(310,270))
    reset1=pygame.image.load('reset1.png').convert()
    screen.blit(reset1,(95,340))
    font=pygame.font.SysFont('timesnewroman',30)
    text=font.render('Play Again',True,BLACK)
    screen.blit(text,(128,355))
    reset2=pygame.image.load('reset2.png').convert()
    screen.blit(reset2,(95,440))
    text=font.render('Exit',True,BLACK)
    screen.blit(text,(170,450))
    pygame.display.update()
    gr='l'
else:
    gr="
return gr

def reset_button():
    while True:
        quit()
        rect1=reset1.get_rect(topleft=(95,340))
        rect2=reset2.get_rect(topleft=(95,440))
        if pygame.mouse.get_pos()[0] in range(rect1.left,rect1.right) and\
pygame.mouse.get_pos()[1] in range(rect1.top,rect1.bottom):
            font=pygame.font.SysFont('timesnewroman',30)
            text=font.render('Play Again',True,GREEN)
            screen.blit(text,(128,355))
            pygame.display.update()
            if pygame.mouse.get_pressed()[0]==1:
                menu()

```

```
    click_newgame()
    x_space=60
    y_space=134
    x,y=60,134
    xb,yb=47,123
    return

    elif pygame.mouse.get_pos()[0] in range(rect2.left,rect2.right) and\
pygame.mouse.get_pos()[1] in range(rect2.top,rect2.bottom):
        font=pygame.font.SysFont('timesnewroman',30)
        text=font.render('Exit',True,GREEN)
        screen.blit(text,(170,450))
        pygame.display.update()
        if pygame.mouse.get_pressed()[0]==1:
            pygame.quit()
            sys.exit()
        else:
            font=pygame.font.SysFont('timesnewroman',30)
            text=font.render('Play Again',True,BLACK)
            screen.blit(text,(128,355))
            text=font.render('Exit',True,BLACK)
            screen.blit(text,(170,450))
            pygame.display.update()

def game_board():
    screen=pygame.display.set_mode((386,700))
    screen.fill(WHITE)
    pygame.display.set_caption('Wordle')
    icon=pygame.image.load('w.jpg')
    pygame.display.set_icon(icon)
```

```
bg=pygame.image.load('tilesfin.jpeg').convert()
screen.blit(bg,(0,0))
pygame.display.update()
```

```
def menu():
```

```
    global BLACK,WHITE,newgame,options,exit1,screen
    screen=pygame.display.set_mode((386,700))
    screen.fill(WHITE)
    pygame.display.set_caption('Wordle')
    icon=pygame.image.load('w.jpg')
    pygame.display.set_icon(icon)
    bg=pygame.image.load('menu.png').convert()
    screen.blit(bg,(0,0))
    font=pygame.font.Font('freesansbold.ttf',50)
    text=font.render('WORDLE',True,BLACK)
    screen.blit(text,(80,60))
    newgame=pygame.image.load('newgame.jpg').convert()
    screen.blit(newgame,(30,170))
    options=pygame.image.load('options.jpg').convert()
    screen.blit(options,(30,290))
    exit1=pygame.image.load('exit.jpg').convert()
    screen.blit(exit1,(30,410))
    pygame.display.update()
```

```
def rules():
```

```
    global Continue,screen
    pygame.display.set_caption('Wordle')
    icon=pygame.image.load('w.jpg')
    pygame.display.set_icon(icon)
```

```

screen=pygame.display.set_mode((386,700))
screen.fill(WHITE)
bg=pygame.image.load('menu.png').convert()
screen.blit(bg,(0,0))
Continue=pygame.image.load('Continue.jpg').convert()
screen.blit(Continue,(30,530))
font=pygame.font.Font('freesansbold.ttf',50)
text=font.render('WORDLE',True,BLACK)
screen.blit(text,(80,40))
font=pygame.font.SysFont('timesnewroman',20)
y=110
with open('rules.txt','r') as F:
    r=F.readlines()
    for k in r[:10]:
        text=font.render(k[:-1],True,BLACK)
        screen.blit(text,(20,y))
        y+=20
    y=410
    rules=pygame.image.load('rules.jpg').convert()
    screen.blit(rules,(50,330))
    for k in r[10:]:
        text=font.render(k[:-1],True,BLACK)
        screen.blit(text,(20,y))
        y+=20
pygame.display.update()

def click_rules():
    global Continue, screen
    rect=Continue.get_rect(topleft=(30,530))

```

```
while True:
    quit()
    if pygame.mouse.get_pos()[0] in range(rect.left,rect.right) and\
pygame.mouse.get_pos()[1] in range(rect.top,rect.bottom):
        Continue=pygame.transform.scale(Continue,(334,78))
        screen.blit(Continue,(28,530))
        pygame.display.update()
        if pygame.mouse.get_pressed()[0]==1:
            menu()
            return
        else:
            rules()

def quit():
    for event in pygame.event.get():
        if event.type==pygame.QUIT:
            pygame.quit()
            sys.exit()

def click_newgame():
    global newgame, exit1, gb, options, new
    time.sleep(1)
    rect1=newgame.get_rect(topleft=(30,170))
    rect2=exit1.get_rect(topleft=(30,410))
    rect3=options.get_rect(topleft=(30,290))
    while True:
        quit()
        if pygame.mouse.get_pos()[0] in range(rect1.left,rect1.right) and\
pygame.mouse.get_pos()[1] in range(rect1.top,rect1.bottom):
```

```
newgame=pygame.transform.scale(newgame,(334,78))
screen.blit(newgame,(28,168))
pygame.display.update()
if pygame.mouse.get_pressed()[0]==1:
    if new==True:
        game_board();gb=True
        indicator()
        return
    else:
        font=pygame.font.SysFont('timesnewroman',30)
        text=font.render('Choose difficulty level',True,BLACK)
        screen.blit(text,(80,120))
elif pygame.mouse.get_pos()[0] in range(rect2.left,rect2.right) and\
pygame.mouse.get_pos()[1] in range(rect2.top,rect2.bottom):
    exit1=pygame.transform.scale(exit1,(334,78))
    screen.blit(exit1,(28,407))
    pygame.display.update()
    if pygame.mouse.get_pressed()[0]==1:
        pygame.quit()
        sys.exit()
elif pygame.mouse.get_pos()[0] in range(rect3.left,rect3.right) and\
pygame.mouse.get_pos()[1] in range(rect3.top,rect3.bottom):
    options=pygame.transform.scale(options,(334,78))
    screen.blit(options,(28,287))
    pygame.display.update()
    if pygame.mouse.get_pressed()[0]==1:
        words()
        time.sleep(1)
        word_click()
```

```
    else:
        menu()

def indicator():
    global S1,S2,S3
    in_x=20
    l_x=27
    for k in range(len(S1)):
        pygame.draw.rect(screen,LIGHTGRAY,pygame.Rect(in_x,500,30,40))
        add_letter(S1[k],l_x,510,BLACK,LIGHTGRAY,20)
        in_x+=36
        l_x+=36
    in_x=35
    l_x=42
    for k in range(len(S2)):
        pygame.draw.rect(screen,LIGHTGRAY,pygame.Rect(in_x,550,30,40))
        add_letter(S2[k],l_x,560,BLACK,LIGHTGRAY,20)
        in_x+=36
        l_x+=36
    in_x=70
    l_x=77
    for k in range(len(S3)):
        pygame.draw.rect(screen,LIGHTGRAY,pygame.Rect(in_x,600,30,40))
        add_letter(S3[k],l_x,610,BLACK,LIGHTGRAY,20)
        in_x+=36
        l_x+=36
    pygame.display.update()

def indicator_color(S,in_x,in_y,l_x,l_y):
```

```
for k in range(len(S)):
    if S[k] in D:
        if D[S[k]]=='green':
            pygame.draw.rect(screen, GREEN, pygame.Rect(in_x, in_y, 30, 40))
            add_letter(S[k], l_x, l_y, WHITE, GREEN, 20)
            in_x+=36
            l_x+=36
        elif D[S[k]]=='yellow':
            pygame.draw.rect(screen, YELLOW, pygame.Rect(in_x, in_y, 30, 40))
            add_letter(S[k], l_x, l_y, WHITE, YELLOW, 20)
            in_x+=36
            l_x+=36
        else:
            pygame.draw.rect(screen, GRAY, pygame.Rect(in_x, in_y, 30, 40))
            add_letter(S[k], l_x, l_y, WHITE, GRAY, 20)
            in_x+=36
            l_x+=36
    else:
        pygame.draw.rect(screen, LIGHTGRAY, pygame.Rect(in_x, in_y, 30, 40))
        add_letter(S[k], l_x, l_y, BLACK, LIGHTGRAY, 20)
        in_x+=36
        l_x+=36
pygame.display.update()
```

BLACK = (0, 0, 0)

GRAY = (127, 127, 127)

WHITE = (255, 255, 255)

RED = (255, 0, 0)

GREEN = (102, 205, 0)


```
BLUE = (0, 0, 255)
YELLOW=(227,207,87)
LIGHTGRAY=(193,205,205)
GOLD=(255,215,0)
S1,S2,S3='QWERTYUIOP','ASDFGHJKL','ZXCVBNM'
newgame=options=exit1=screen=Continue=""
reset1=reset2=""
D={}
D1={}
word=""
easy=medium=hard=new=""
gb=""
c=0
tc=ws=0
L=[]
x_space=60
y_space=134
x,y=60,134
xb,yb=47,123
green=0

while True:
    rules()
    click_rules()
    click_newgame()
    while gb:
        for event in pygame.event.get():
            if event.type==pygame.QUIT:
                pygame.quit()
```

```

sys.exit()
if event.type==pygame.KEYDOWN:
    if pygame.key.name(event.key).isalpha():
        if len(pygame.key.name(event.key))==1 and c<5:
            key=pygame.key.name(event.key).upper()
            add_letter(key,x_space,y_space,BLACK,WHITE,32)
            x_space+=60
            c+=1
            L+=[key]

    if event.key==pygame.K_RETURN:
        if c==5:
            for k in word:
                if k not in D1:
                    D1[k]=1
                else:
                    D1[k]+=1
            colour(word,L)
            tc+=1
            D1={}
            pygame.draw.rect(screen,WHITE,pygame.Rect(60,60,300,51))
            pygame.display.update()
            indicator_color(S1,20,500,27,510)
            indicator_color(S2,35,550,42,560)
            indicator_color(S3,70,600,77,610)
            time.sleep(1)
            result()
            if result()=='w':
                ws+=1

```

```

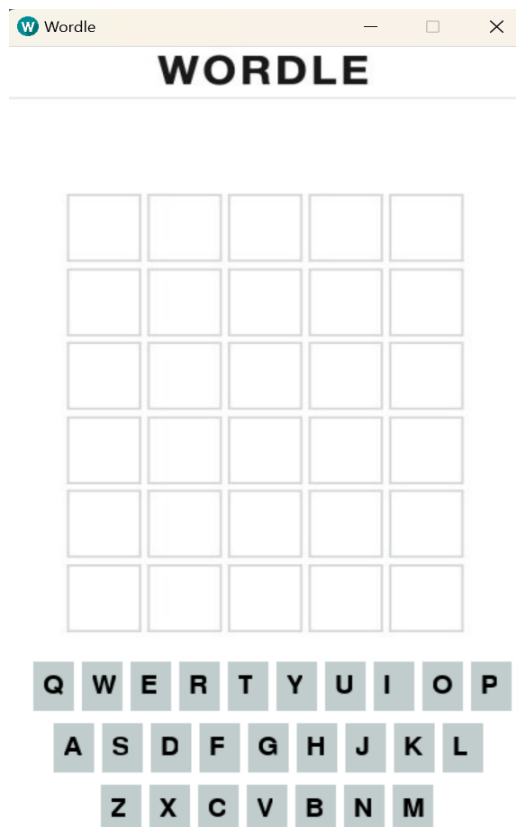
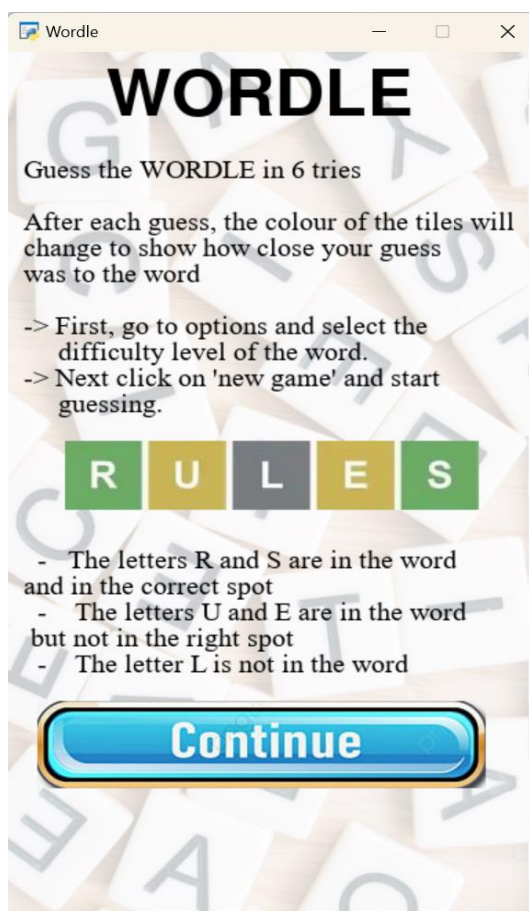
    if result() in ('w','l'):
        x_space=60
        y_space=76
        x,y=60,134
        xb,yb=47,123
        tc=0
        D={ }
        D1={ }
        reset_button()
    x_space=60
    y_space+=58
    L=[]
    c=0
    green=0

else:
    pygame.draw.rect(screen,WHITE,pygame.Rect(60,60,100,51))
    font=pygame.font.Font('freesansbold.ttf',20)
    text=font.render('The word must contain atleast',True,BLACK)
    screen.blit(text,(60,60))
    text=font.render('          5 letters',True,BLACK)
    screen.blit(text,(60,80))
    pygame.display.update()
if event.key==pygame.K_BACKSPACE:
    if L==[]:
        pygame.draw.rect(screen,WHITE,pygame.Rect(60,60,100,51))
        font=pygame.font.Font('freesansbold.ttf',20)
        text=font.render('There are no letters entered',True,BLACK)
        screen.blit(text,(60,60))

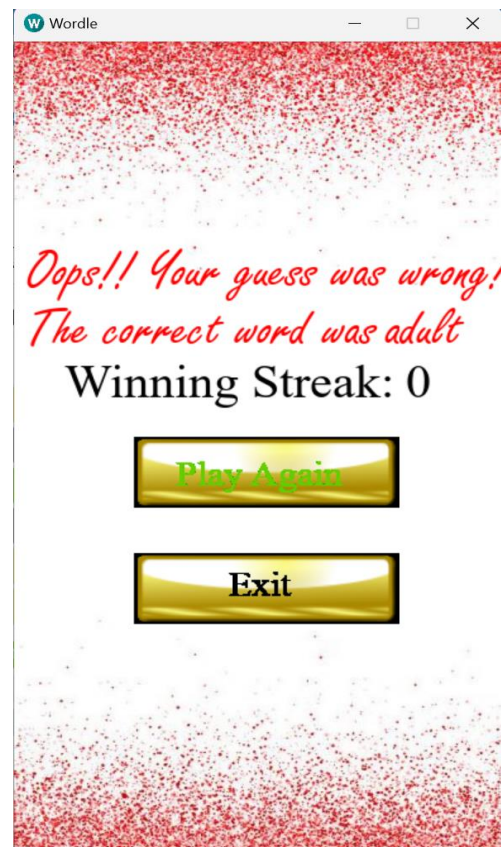
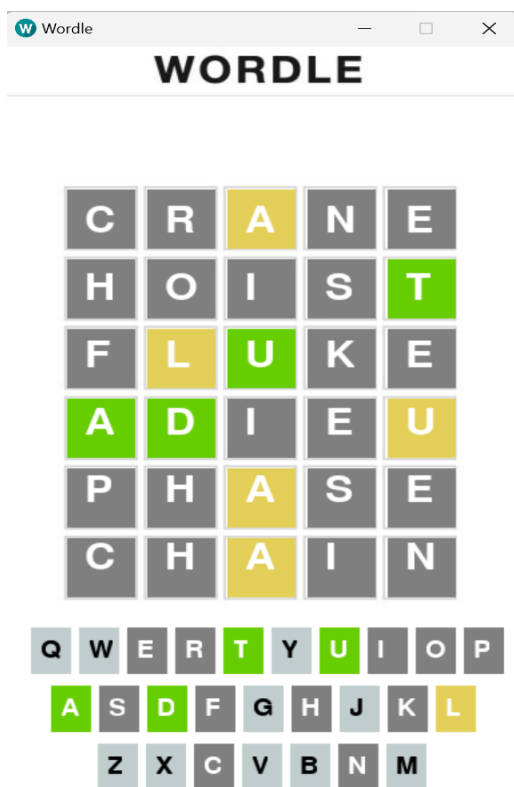
```

```
pygame.display.update()  
else:  
    x_space-=60  
    delete(x_space,y_space)  
    L.pop()  
    c-=1
```

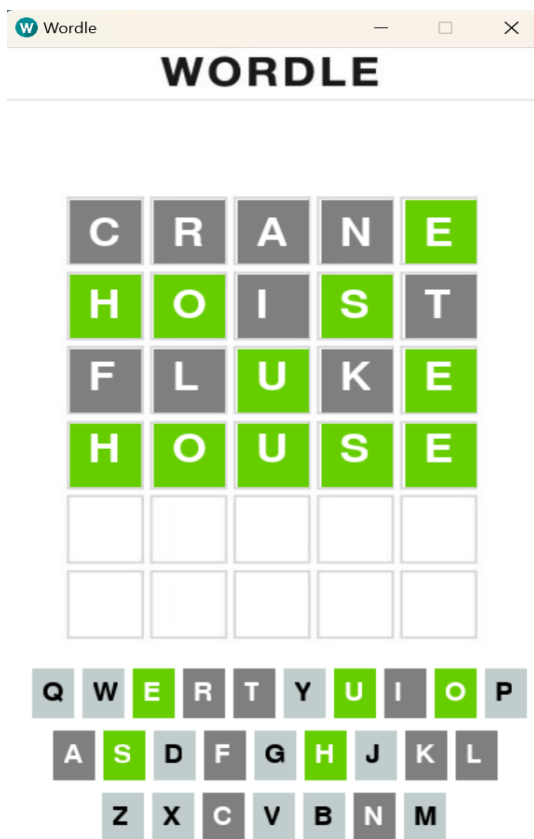
SAMPLE OUPTPUTS



GAME STATUS- LOST



GAME STATUS – WON



BIBLIOGRAPHY

- Class 12 computer science with python by Sumita Arora
- <https://www.nytimes.com/games/wordle/index>
- Beginning game development with python and pygame by Will McGugan
- <https://www.geeksforgeeks.org/>
- <https://github.com/>