Phase 2 Project list

Phase 2 Projects

Domain : Python Development Intern

Normal Task

3. Typing speed Test

SOURCE CODE:-

import pygame

from pygame.locals import \*

import sys

import time

import random

# 750 x 500

class Game:

def \_\_init\_\_(self):

self.w=750

self.h=500

self.reset=True

self.active = False

self.input\_text=''

self.word = ''

self.time\_start = 0

self.total\_time = 0

self.accuracy = '0%'

self.results = 'Time:0 Accuracy:0 % Wpm:0 '

self.wpm = 0

self.end = False

self.HEAD\_C = (255,213,102)

self.TEXT\_C = (240,240,240)

self.RESULT\_C = (255,70,70)

pygame.init()

self.open\_img = pygame.image.load('type-speed-open.png')

self.open\_img = pygame.transform.scale(self.open\_img, (self.w,self.h))

self.bg = pygame.image.load('background.jpg')

self.bg = pygame.transform.scale(self.bg, (500,750))

self.screen = pygame.display.set\_mode((self.w,self.h))

pygame.display.set\_caption('Type Speed test')

def draw\_text(self, screen, msg, y ,fsize, color):

font = pygame.font.Font(None, fsize)

text = font.render(msg, 1,color)

text\_rect = text.get\_rect(center=(self.w/2, y))

screen.blit(text, text\_rect)

pygame.display.update()

def get\_sentence(self):

f = open('sentences.txt').read()

sentences = f.split('\n')

sentence = random.choice(sentences)

return sentence

def show\_results(self, screen):

if(not self.end):

#Calculate time

self.total\_time = time.time() - self.time\_start

#Calculate accuracy

count = 0

for i,c in enumerate(self.word):

try:

if self.input\_text[i] == c:

count += 1

except:

pass

self.accuracy = count/len(self.word)\*100

#Calculate words per minute

self.wpm = len(self.input\_text)\*60/(5\*self.total\_time)

self.end = True

print(self.total\_time)

self.results = 'Time:'+str(round(self.total\_time)) +" secs Accuracy:"+ str(round(self.accuracy)) + "%" + ' Wpm: ' + str(round(self.wpm))

# draw icon image

self.time\_img = pygame.image.load('icon.png')

self.time\_img = pygame.transform.scale(self.time\_img, (150,150))

#screen.blit(self.time\_img, (80,320))

screen.blit(self.time\_img, (self.w/2-75,self.h-140))

self.draw\_text(screen,"Reset", self.h - 70, 26, (100,100,100))

print(self.results)

pygame.display.update()

def run(self):

self.reset\_game()

self.running=True

while(self.running):

clock = pygame.time.Clock()

self.screen.fill((0,0,0), (50,250,650,50))

pygame.draw.rect(self.screen,self.HEAD\_C, (50,250,650,50), 2)

# update the text of user input

self.draw\_text(self.screen, self.input\_text, 274, 26,(250,250,250))

pygame.display.update()

for event in pygame.event.get():

if event.type == QUIT:

self.running = False

sys.exit()

elif event.type == pygame.MOUSEBUTTONUP:

x,y = pygame.mouse.get\_pos()

# position of input box

if(x>=50 and x<=650 and y>=250 and y<=300):

self.active = True

self.input\_text = ''

self.time\_start = time.time()

# position of reset box

if(x>=310 and x<=510 and y>=390 and self.end):

self.reset\_game()

x,y = pygame.mouse.get\_pos()

elif event.type == pygame.KEYDOWN:

if self.active and not self.end:

if event.key == pygame.K\_RETURN:

print(self.input\_text)

self.show\_results(self.screen)

print(self.results)

self.draw\_text(self.screen, self.results,350, 28, self.RESULT\_C)

self.end = True

elif event.key == pygame.K\_BACKSPACE:

self.input\_text = self.input\_text[:-1]

else:

try:

self.input\_text += event.unicode

except:

pass

pygame.display.update()

clock.tick(60)

def reset\_game(self):

self.screen.blit(self.open\_img, (0,0))

pygame.display.update()

time.sleep(1)

self.reset=False

self.end = False

self.input\_text=''

self.word = ''

self.time\_start = 0

self.total\_time = 0

self.wpm = 0

# Get random sentence

self.word = self.get\_sentence()

if (not self.word): self.reset\_game()

#drawing heading

self.screen.fill((0,0,0))

self.screen.blit(self.bg,(0,0))

msg = "Typing Speed Test"

self.draw\_text(self.screen, msg,80, 80,self.HEAD\_C)

# draw the rectangle for input box

pygame.draw.rect(self.screen,(255,192,25), (50,250,650,50), 2)

# draw the sentence string

self.draw\_text(self.screen, self.word,200, 28,self.TEXT\_C)

pygame.display.update()

Game().run()

Golden Task

4. Voice Recorder

Source code:-

# import required libraries

import sounddevice as sd

from scipy.io.wavfile import write

import wavio as wv

# Sampling frequency

freq = 44100

# Recording duration

duration = 5

# Start recorder with the given values

# of duration and sample frequency

recording = sd.rec(int(duration \* freq),

samplerate=freq, channels=2)

# Record audio for the given number of seconds

sd.wait()

# This will convert the NumPy array to an audio

# file with the given sampling frequency

write("recording0.wav", freq, recording)

# Convert the NumPy array to audio file

wv.write("recording1.wav", recording, freq, sampwidth=2)