1. BACKGROUND AND EVENT

import pygame

pygame.init()

# background/screen

screen\_width = 626

screen\_height = 417

display\_output = [screen\_width,screen\_height]

screen = pygame.display.set\_mode((display\_output),pygame.RESIZABLE)

background = pygame.image.load('images/bg\_1.jpg').convert()

# display name

pygame.display.set\_caption("Collect the Trash")

play = True

# 1 event screen/play

def check\_for\_event():

global play

for event in pygame.event.get():

if event.type == pygame.QUIT:

play = False

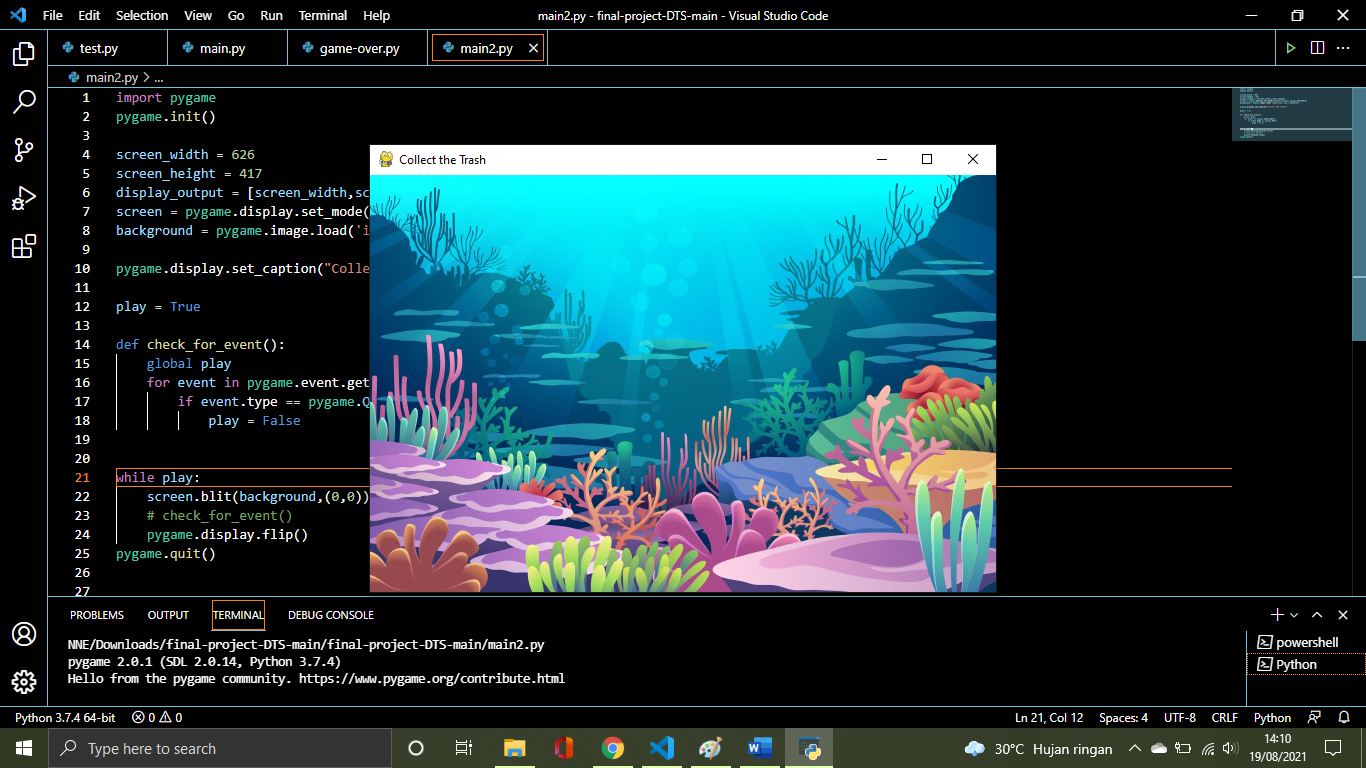
while play:

screen.blit(background,(0,0))

check\_for\_event()

pygame.display.flip()

pygame.quit()



1. INITIALIZED PLASTIC AND PUT RANDOM IN SCREEN WIDTH

another text..

from random import randint

another text..

plastic = pygame.image.load('images/plastic\_raw.png').convert\_alpha()

x = randint(0, screen\_width)

y = 0

radius = 35 # plastic width

plastic\_rep\_x = x + 50 # keep the plastic in range x

plastic\_rep\_y = y + 50 # keep the p;astic in range y

another text..

def show\_images():

    screen.blit(plastic, (x,y))

loop play

while play:

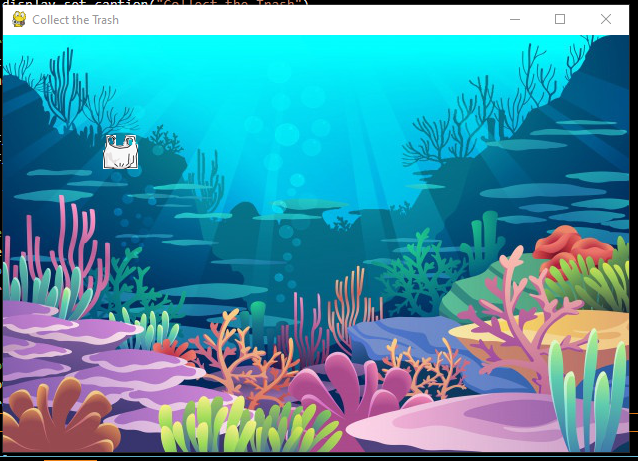
    screen.blit(background,(0,0))

    check\_for\_event()

    show\_images()

    pygame.display.flip()

pygame.quit()



1. INITIALIZED PLASTIC FALL

Initialized speed

speed = 5

function position plastic

def update\_pos\_plastic():

    global y, plastic\_rep\_x, plastic\_rep\_y

    y += speed

    plastic\_rep\_x = x + 50

    plastic\_rep\_y = y + 50

loop play

while play:

    screen.blit(background,(0,0))

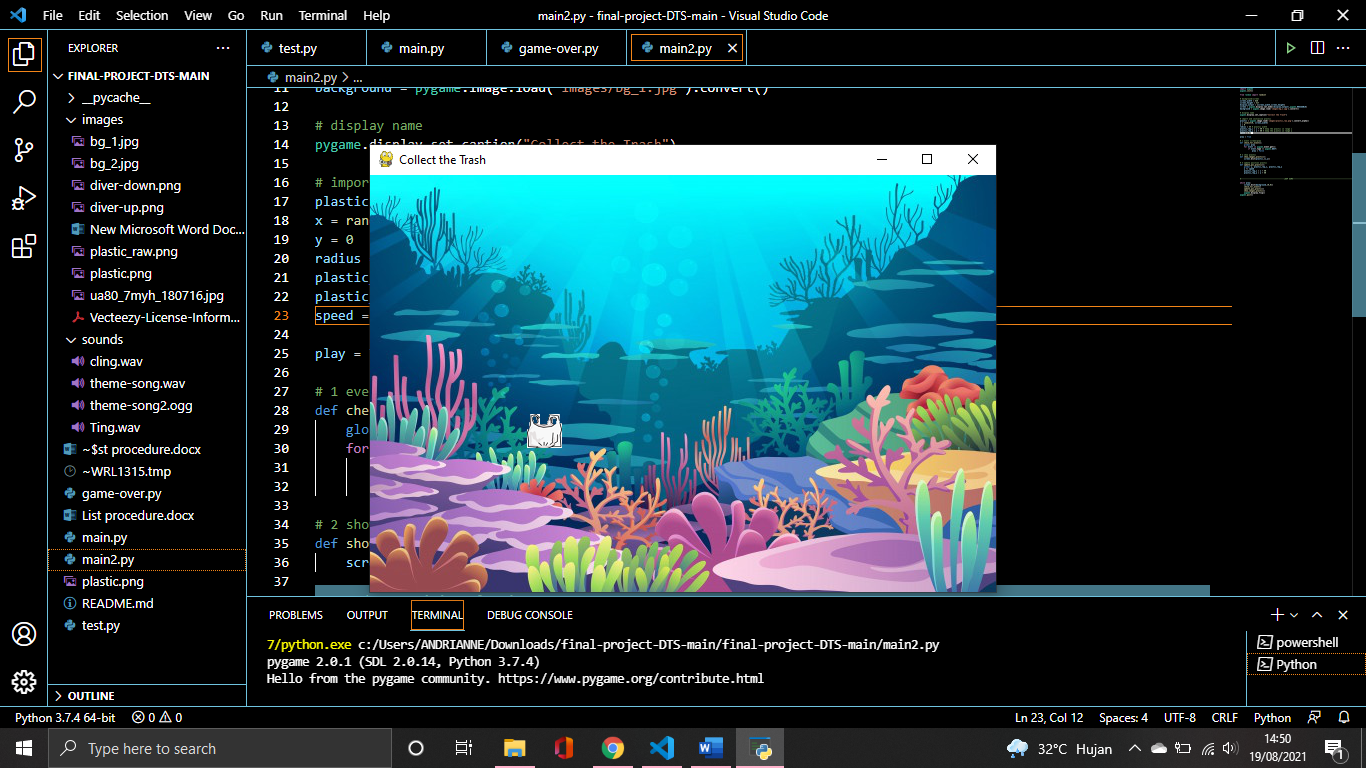
    check\_for\_event()

    update\_pos\_plastic()

    show\_images\_plastic()

    pygame.display.flip()

pygame.quit()



1. SET THE CLOCK SO PLASTIC FALL SLOW DOWN

Set the clock

clock = pygame.time.Clock()

loop play

while play:

    clock.tick(60) #set 60

    screen.blit(background,(0,0))

    check\_for\_event()

    update\_pos\_plastic()

    show\_images\_plastic()

    pygame.display.flip()

pygame.quit()

1. INITIALIZES PLASTIC POSITION AND ADDING BORDER

Initialise new random plastic down

def initialise\_plastic():

    global x, y

    if y > screen\_height - radius:

        y = 0

        x = randint(0, screen\_width)

call the function after update\_position\_plastic

def update\_pos\_plastic():

def update\_pos\_plastic():

    global y, plastic\_rep\_x, plastic\_rep\_y

    y += speed

    plastic\_rep\_x = x + 50

    plastic\_rep\_y = y + 50

    initialise\_plastic()

call enforce border into loop play

while play:

    clock.tick(60)

    screen.blit(background,(0,0))

    check\_for\_event()

    update\_pos\_plastic()

    enforce\_border()

    show\_images\_plastic()

    pygame.display.flip()

pygame.quit()

1. IMPORT PLAYER AND POSITION

Import images and initialize position and region player

player = pygame.image.load("images/diver2.png").convert\_alpha()

player\_width = 175

player\_pos\_x = 200

player\_pos\_y = 450

player\_speed = 8

player\_rep\_x = [player\_pos\_x + 31, player\_pos\_x + 94] # region to score

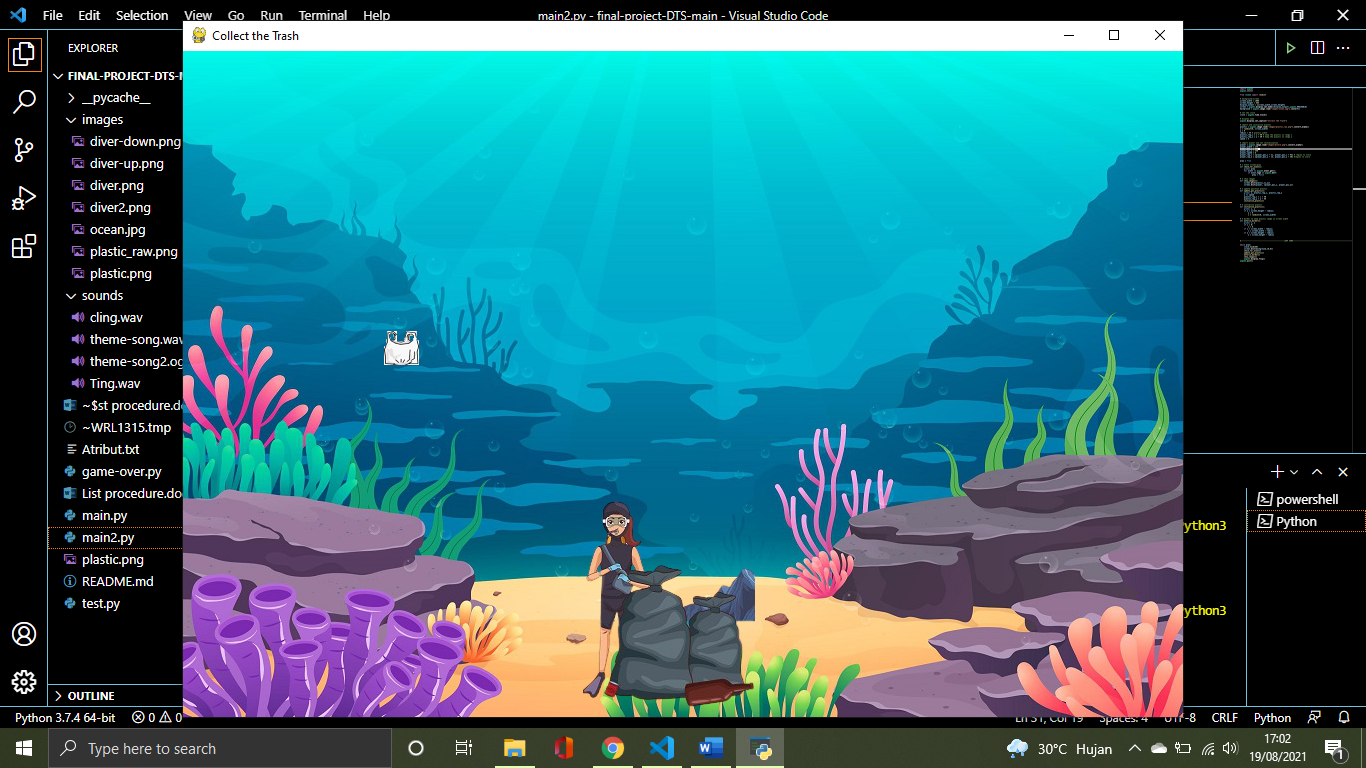
player\_rep\_y = [player\_pos\_y + 14, player\_pos\_y + 20] #region to score

call player into show\_images function

def show\_images():

    screen.blit(plastic,(x,y))

    screen.blit(player, (player\_pos\_x, player\_pos\_y))



1. MAKE THE DIVER SLIDE RIGHT-LEFT

With Left and Right Keys, move the player to the left or right

def check\_for\_event():

    global play, player\_pos\_x, player\_pos\_y

    for event in pygame.event.get():

        if event.type == pygame.QUIT:

            play = False

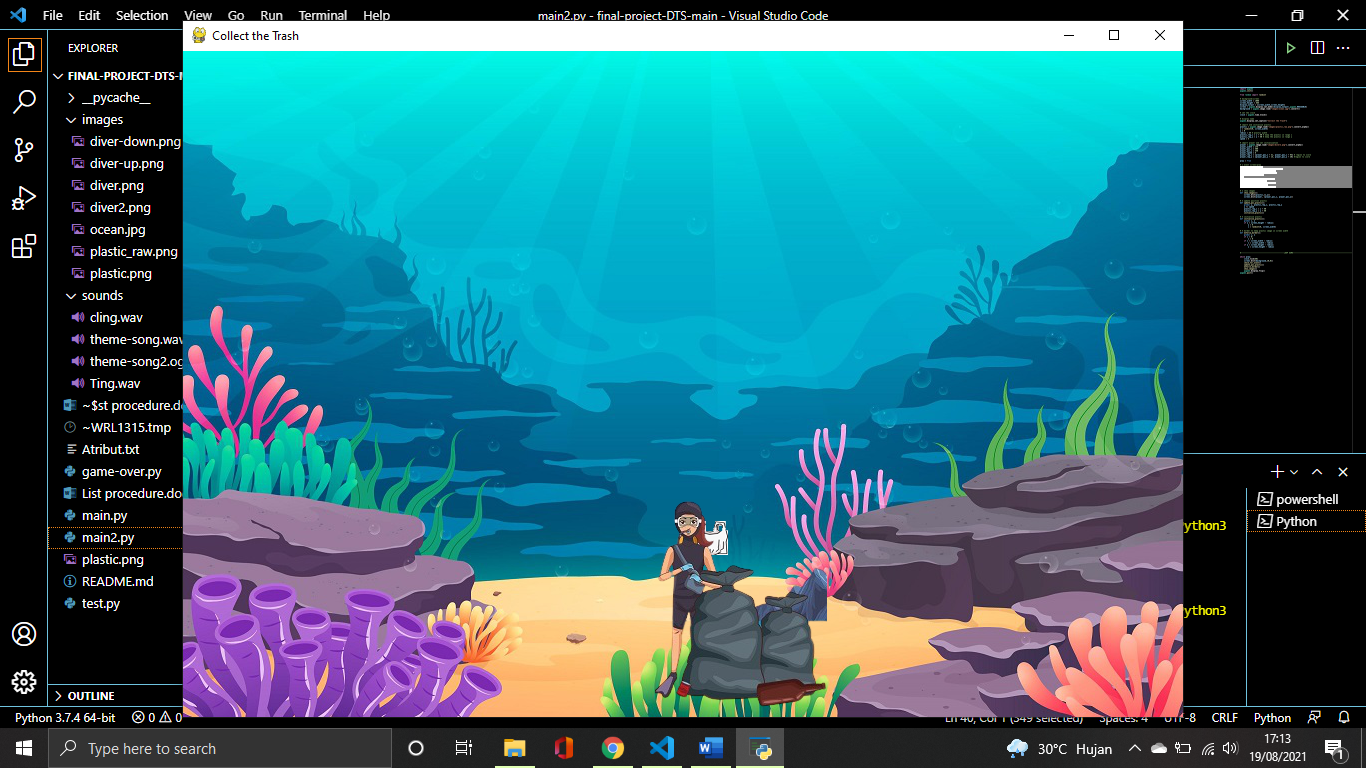
    keys = pygame.key.get\_pressed()

    if keys[pygame.K\_LEFT]:

        player\_pos\_x -= player\_speed

    if keys[pygame.K\_RIGHT]:

        player\_pos\_x += player\_speed



1. BORDER FOR DIVER

Add code into enforce\_border, if the player less than or more than screen width

def enforce\_border():

    global x, y, player\_pos\_x, player\_pos\_y

    if x < 0:

        x = 0

    if x > screen\_width - radius:

        x = screen\_width - radius

    if y > screen\_height - radius:

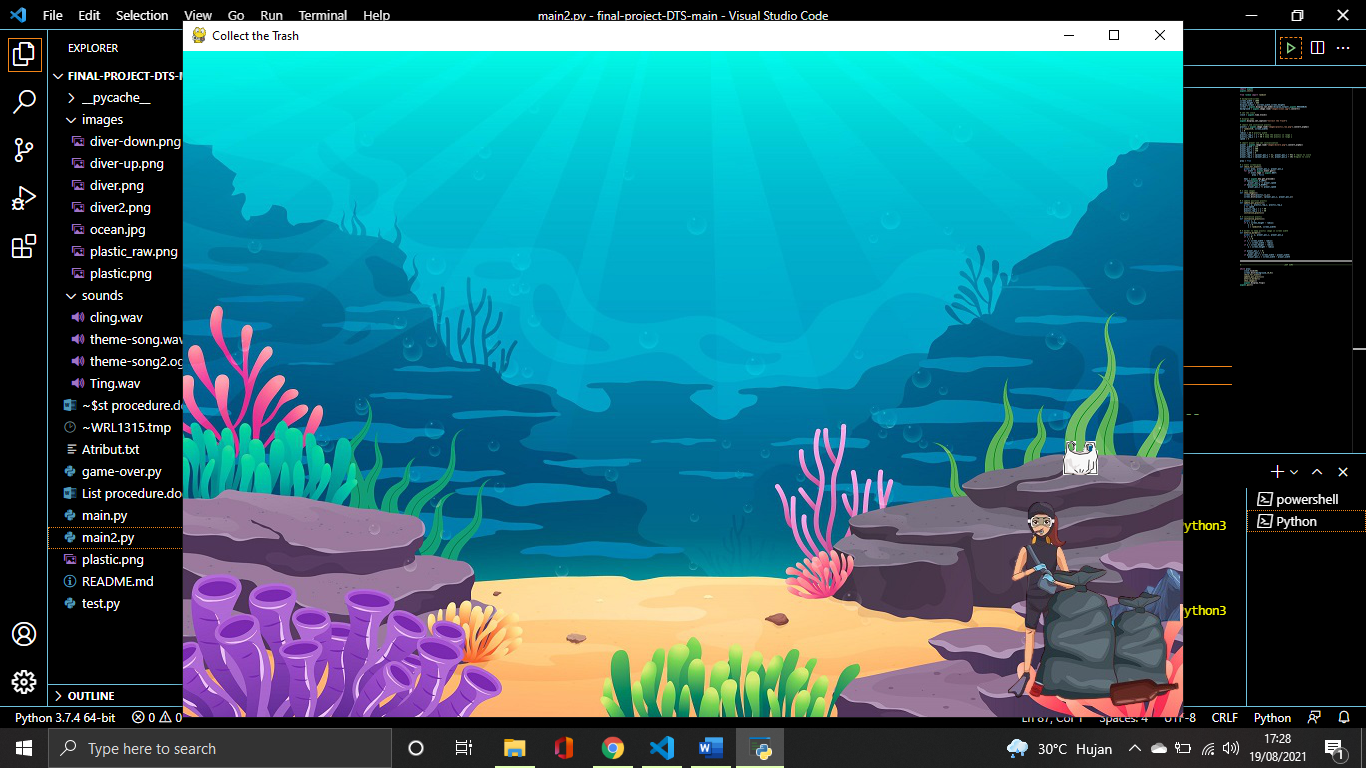
        y = screen\_height - radius

    if player\_pos\_x < 0:

        player\_pos\_x = 0

    if player\_pos\_x > screen\_width - player\_width:

        player\_pos\_x = screen\_width - player\_width



1. COUNT SCORE OR RESET SCORE

Add function to declare plastic fall into player region

def check\_for\_score():

    global score

    if plastic\_rep\_x in range(player\_rep\_x[0], player\_rep\_x[1]) and plastic\_rep\_y in range (player\_rep\_y[0], player\_rep\_y[1]):

        score += 1

    elif plastic\_rep\_y in range(player\_rep\_y[0], player\_rep\_y[1]) and plastic\_rep\_x in range (player\_rep\_x[0], player\_rep\_x[1]):

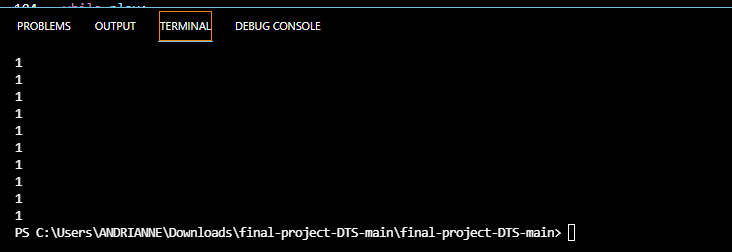
        score = 0

show score

def show\_score():

    print(score)

Output print score



1. DISPLAY SCORE

Initialization text score

black = (0, 0, 0)

text\_x = 15

text\_y = 15

font = pygame.font.Font("freesansbold.ttf", 40)

Add code into show\_score function

def show\_score():

    score\_display = font.render("Score " + str(score), True, black)

    screen.blit(score\_display, (text\_x, text\_y))

Call function in loop play

while play:

    clock.tick(60)

    screen.blit(background,(0,0))

    check\_for\_event()

    update\_pos\_plastic()

    enforce\_border()

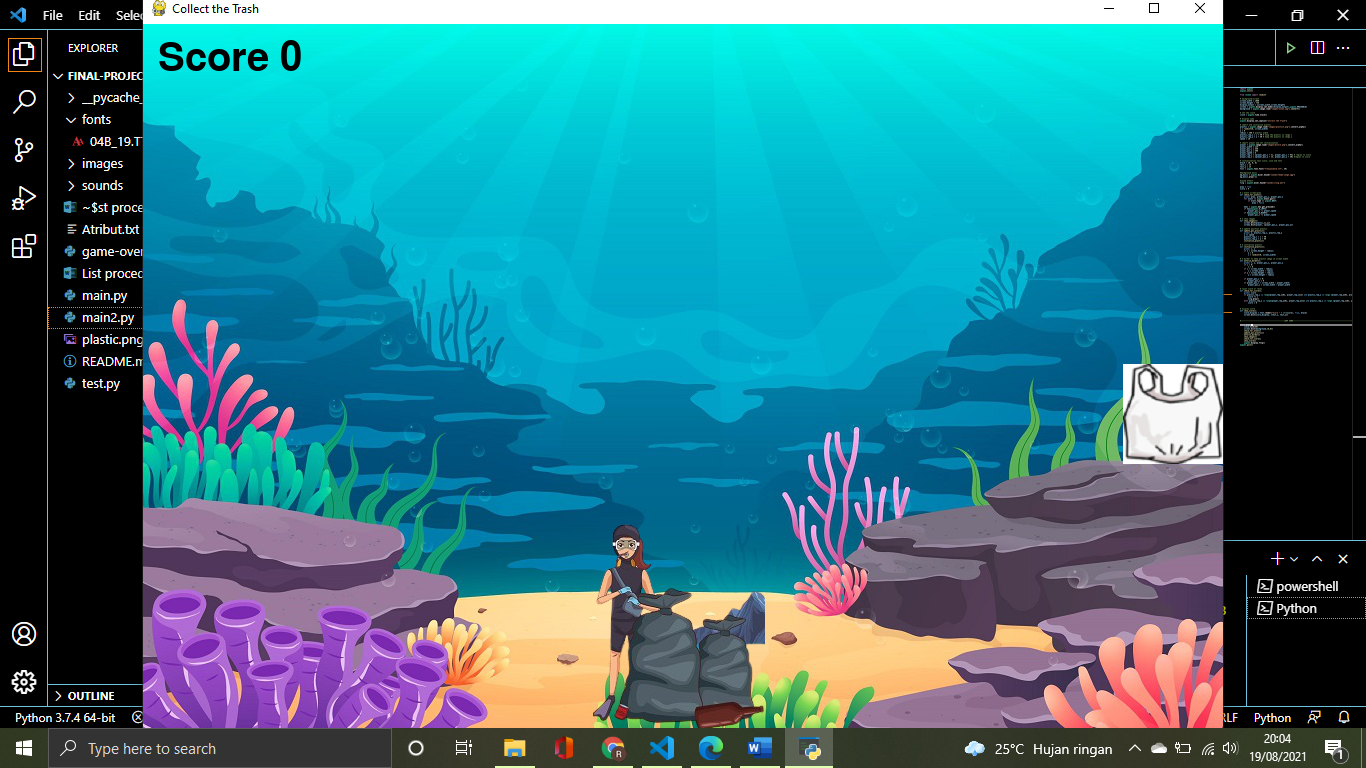
    show\_images()

    check\_for\_score()

    show\_score()

    pygame.display.flip()

pygame.quit()



1. ADD SOUND THEME AND SOUND EFFECT

Initialization background music and sound effect

bg\_music = pygame.mixer.Sound("sounds/theme-song2.ogg")

bg\_music.play(-1)

ting = pygame.mixer.Sound("sounds/cling.wav")

Add code into check\_for\_score funtion

def check\_for\_score():

    global score

    if plastic\_rep\_x in range(player\_rep\_x[0], player\_rep\_x[1]) and plastic\_rep\_y in range(player\_rep\_y[0], player\_rep\_y[1]):

        score += 1

        ting.play() #sound effect

    elif plastic\_rep\_y in range(player\_rep\_y[0], player\_rep\_y[1]) and plastic\_rep\_y not in range(player\_rep\_x[0], player\_rep\_x[1]):

        score = 0