NAME:K.VENKATAKIRAN

REG NO:19BEC0583

From VIT university

Flappy\_bird\_game\_using\_python

CODE:

import random # For generating random numbers

import sys # We will use sys.exit to exit the program

import pygame

from pygame.locals import \* # Basic pygame imports

# Global Variables for the game

FPS = 40  # number of times the frame will be rendered in 1 sec

SCREENWIDTH = 289   # defining width

SCREENHEIGHT = 511  # defining height

GROUNDY = SCREENHEIGHT \* 0.8

SCREEN = pygame.display.set\_mode((SCREENWIDTH, SCREENHEIGHT))  # sending height and

GAME\_SPRITES = {}

PLAYER = 'gallery/sprites/bird.png'

BACKGROUND = 'gallery/sprites/background.png'

PIPE = 'gallery/sprites/pipe.png'

GAME\_SPRITES['message'] =pygame.image.load('gallery/sprites/message.png').convert\_alpha()

GAME\_SPRITES['base'] =pygame.image.load('gallery/sprites/base.png').convert\_alpha()

GAME\_SPRITES['background'] = pygame.image.load(BACKGROUND).convert()

GAME\_SPRITES['player'] = pygame.image.load(PLAYER).convert\_alpha()

GAME\_SPRITES['pipe'] =(pygame.transform.rotate(pygame.image.load( PIPE).convert\_alpha(), 180), pygame.image.load(PIPE).convert\_alpha())

pygame.init() # Initialize all pygame's modules

FPSCLOCK = pygame.time.Clock()

pygame.display.set\_caption('Flappy Bird CSI')

def welcomeScreen():

    """

    Shows welcome images on the screen

    """

    playerx = int(SCREENWIDTH/5)

    playery = int((SCREENHEIGHT - GAME\_SPRITES['player'].get\_height())/2)

    messagex = int((SCREENWIDTH - GAME\_SPRITES['message'].get\_width())/2)

    messagey = int(SCREENHEIGHT\*0.13)

    basex = 0

    #Game

    while True:

        for event in pygame.event.get():

            # if user clicks on cross button, close the game

            if event.type == QUIT or (event.type==KEYDOWN and event.key == K\_ESCAPE):

                pygame.quit()

                sys.exit()

            # If the user presses space or up key, start the game for them

            if event.type==KEYDOWN and (event.key==K\_SPACE or event.key == K\_UP):

                return

            else:

                SCREEN.blit(GAME\_SPRITES['background'], (0, 0))

                SCREEN.blit(GAME\_SPRITES['player'], (playerx, playery))

                SCREEN.blit(GAME\_SPRITES['message'], (messagex,messagey ))

                SCREEN.blit(GAME\_SPRITES['base'], (basex, GROUNDY))

                pygame.display.update()

                FPSCLOCK.tick(FPS)

def mainGame():

    playerx = int(SCREENWIDTH/5)

    playery = int(SCREENWIDTH/2)

    basex = 0

    # Create 2 pipes for blitting on the screen

    newPipe1 = getRandomPipe()

    newPipe2 = getRandomPipe()

    # my List of upper pipes

    upperPipes = [

        {'x': SCREENWIDTH+200, 'y':newPipe1[0]['y']},

        {'x': SCREENWIDTH+200+(SCREENWIDTH/2), 'y':newPipe2[0]['y']},

    ]

    # my List of lower pipes

    lowerPipes = [

        {'x': SCREENWIDTH+200, 'y':newPipe1[1]['y']},

        {'x': SCREENWIDTH+200+(SCREENWIDTH/2), 'y':newPipe2[1]['y']},

    ]

    pipeVelX = -4

    playerVelY = -9

    playerMaxVelY = 10

    playerMinVelY = -8

    playerAccY = 1

    playerFlapAccv = -8 # velocity while flapping

    while True:

        for event in pygame.event.get():

            if event.type == QUIT or (event.type == KEYDOWN and event.key == K\_ESCAPE):

                pygame.quit()

                sys.exit()

            if event.type == KEYDOWN and (event.key == K\_SPACE or event.key == K\_UP):

                if playery > 0: #Player is in the screen

                    playerVelY = playerFlapAccv

        if playerVelY <playerMaxVelY:

            playerVelY += playerAccY

            print(playerVelY)

        playerHeight = GAME\_SPRITES['player'].get\_height()

        playery = playery + min(playerVelY, GROUNDY - playery - playerHeight)

        # move pipes to the left

        for upperPipe , lowerPipe in zip(upperPipes, lowerPipes):

            upperPipe['x'] += pipeVelX

            lowerPipe['x'] += pipeVelX

        # Add a new pipe when the first is about to cross the leftmost part of the screen

        if 0<upperPipes[0]['x']<5:

            newpipe = getRandomPipe()

            upperPipes.append(newpipe[0])

            lowerPipes.append(newpipe[1])

        # if the pipe is out of the screen, remove it

        if upperPipes[0]['x'] < -GAME\_SPRITES['pipe'][0].get\_width():

            upperPipes.pop(0)

            lowerPipes.pop(0)

        # Lets blit our sprites now

        SCREEN.blit(GAME\_SPRITES['background'], (0, 0))

        for upperPipe, lowerPipe in zip(upperPipes, lowerPipes):

            SCREEN.blit(GAME\_SPRITES['pipe'][0], (upperPipe['x'], upperPipe['y']))

            SCREEN.blit(GAME\_SPRITES['pipe'][1], (lowerPipe['x'], lowerPipe['y']))

        SCREEN.blit(GAME\_SPRITES['base'], (basex, GROUNDY))

        SCREEN.blit(GAME\_SPRITES['player'], (playerx, playery))

        pygame.display.update()

        FPSCLOCK.tick(FPS)

def getRandomPipe():

    pipeHeight=GAME\_SPRITES['pipe'][0].get\_height()

    fixpoint=int(SCREENHEIGHT/3)

    y2=random.randint(int(fixpoint+0.2\*fixpoint),int(SCREENHEIGHT-GAME\_SPRITES['base'].get\_height()-0.5\*fixpoint))

    y1=pipeHeight-y2+100

    pipex=SCREENWIDTH+10

    pipe=[

        {'x':pipex,'y':-y1},#Upper Pipe

        {'x':pipex,'y':y2}#lowepipe

    ]

    return pipe

while True:

    welcomeScreen() # Shows welcome screen to the user until he presses a button

    mainGame()

