import pygame

from pygame.locals import\*

import random

pygame.init()

clock=pygame.time.Clock()

fps=60

screen\_width=500

screen\_height=265

screen=pygame.display.set\_mode((screen\_width,screen\_height))

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

#define font

font=pygame.font.SysFont('Bauhaus 93',60)

#define color

white=(255,255,255)

#define game variables

ground\_scroll=0

scroll\_speed=4

flying=False

game\_over=False

pipe\_gap=200

pipe\_frequency=1500#millisecs

last\_pipe=pygame.time.get\_ticks()-pipe\_frequency

score=0

pass\_pipe=False

#load images

bg=pygame.image.load("C:\\Users\\ROSHINI\\OneDrive\\Documents\\bg1.png")

ground\_img=pygame.image.load("C:\\Users\\ROSHINI\\Downloads\\s-0.png")

button\_img=pygame.image.load("C:\\Users\\ROSHINI\\OneDrive\\Documents\\restart (1).png")

def draw\_text(text,font,text\_col,x,y):

img=font.render(text,True,text\_col)

screen.blit(img,(x,y))

def reset\_game():

pipe\_group.empty()

flappy.rect.x=100

flappy.rect.y=int(screen\_height/2)

score=0

return score

class Bird(pygame.sprite.Sprite):

def \_\_init\_\_ (self,x,y):

pygame.sprite.Sprite. \_\_init\_\_(self)

self.images=[]

self.index=0

self.counter=0

for num in range(0,3):

img=pygame.image.load(f'C:\\Users\\ROSHINI\\Downloads\\red{num}.png')

self.images.append(img)

self.image=self.images[self.index]

self.rect=self.image.get\_rect()

self.rect.center=[x,y]

self.vel=0

self.clicked = False

def update(self):

if flying==True:

self.vel+=0.5

if self.vel>8:

self.vel=8

if self.rect.bottom<250:

self.rect.y+=int(self.vel)

if game\_over ==False:

#jump

if pygame.mouse.get\_pressed()[0] == 1 and self.clicked==False:

self.clicked=True

self.vel=-10

if pygame.mouse.get\_pressed()[0] == 0:

self.clicked=False

#handle the animation

self.counter+=1

flap\_cooldown=5

if self.counter> flap\_cooldown:

self.counter=0

self.index+=1

if self.index >= len(self.images):

self.index=0

self.image=self.images[self.index]

#rotate

self.image=pygame.transform.rotate(self.images[self.index],self.vel\*-2)

else:

self.image=pygame.transform.rotate(self.images[self.index],-90)

class Pipe(pygame.sprite.Sprite):

def \_\_init\_\_(self, x, y, position):

pygame.sprite.Sprite.\_\_init\_\_(self)

self.image = pygame.image.load("C:\\Users\\ROSHINI\\OneDrive\\Documents\\peak1.png")

self.rect = self.image.get\_rect()

#position 1 is from the top, -1 is from the bottom

if position == 1:

self.image = pygame.transform.flip(self.image, False, True)

self.rect.bottomleft = [x, y - int(pipe\_gap / 2)]

if position == -1:

self.rect.topleft = [x, y + int(pipe\_gap / 2)]

def update(self):

self.rect.x -= scroll\_speed

if self.rect.right < 0:

self.kill()

class Button():

def \_\_init\_\_(self,x,y,image):

self.image=image

self.rect=self.image.get\_rect()

self.rect.topleft=(x,y)

def draw(self):

action=False

#get mouse position

pos=pygame.mouse.get\_pos()

#check if is mouse is over the button

if self.rect.collidepoint(pos):

if pygame.mouse.get\_pressed()[0] == 1:

action=True

#draw button

screen.blit(self.image,(self.rect.x,self.rect.y))

return action

bird\_group=pygame.sprite.Group()

pipe\_group=pygame.sprite.Group()

flappy= Bird(100, int (screen\_height / 2))

bird\_group.add(flappy)

#create restart button

button=Button(screen\_width//2-50,screen\_height//2-100,button\_img)

run = True

while run:

clock.tick(fps)

#draw backgroung

screen.blit(bg,(0 , 1))

bird\_group.draw (screen)

bird\_group.update()

pipe\_group.draw (screen)

#draw the ground

screen.blit(ground\_img,(ground\_scroll,250))

#check the score

if len(pipe\_group) > 0:

if bird\_group.sprites()[0].rect.left > pipe\_group.sprites()[0].rect.left\

and bird\_group.sprites()[0].rect.right < pipe\_group.sprites()[0].rect.right\

and pass\_pipe==False:

pass\_pipe=True

if pass\_pipe==True:

if bird\_group.sprites()[0].rect.left > pipe\_group.sprites()[0].rect.right:

score +=1

pass\_pipe=False

draw\_text(str(score),font,white,int(screen\_width/2),20)

#look for collisions

if pygame.sprite.groupcollide(bird\_group,pipe\_group,False,False) or flappy.rect.top<0:

game\_over=True

#check if bird has hit the ground

if flappy.rect.bottom >= 250:

game\_over=True

flying=False

if game\_over==False and flying==True:

#generate new pipes

time\_now=pygame.time.get\_ticks()

if time\_now-last\_pipe>pipe\_frequency:

pipe\_height=random.randint(-100,100)

btm\_pipe=Pipe(screen\_width,int (screen\_height / 2)+pipe\_height,-1)

top\_pipe=Pipe(screen\_width,int (screen\_height / 2)+pipe\_height,1)

pipe\_group.add(btm\_pipe)

pipe\_group.add(top\_pipe)

last\_pipe=time\_now

#draw and scroll the ground

ground\_scroll -= scroll\_speed

if abs(ground\_scroll) > 55:

ground\_scroll = 0

pipe\_group.update()

#check for game over and restart

if game\_over==True:

if button.draw()==True:

game\_over=False

reset\_game()

score=reset\_game()

for event in pygame.event.get():

if event.type == pygame.QUIT:

run = False

if event.type==pygame.MOUSEBUTTONDOWN and flying==False and game\_over==False:

flying=True

pygame.display.update()

pygame.quit()

IMAGES

   



 