import pygame

import os

import random

TELA\_LARGURA = 550

TELA\_ALTURA = 800

IMAGEM\_CANO = pygame.transform.scale2x(pygame.image.load(os.path.join('imgs', 'pipe.png')))

IMAGEM\_CANO\_VANILLA = pygame.transform.scale2x(pygame.image.load(os.path.join('imgs', 'Vanilla\_pipe.png')))

IMAGEM\_CANO\_AZUL = pygame.transform.scale2x(pygame.image.load(os.path.join('imgs', 'Blue\_pipe.png')))

IMAGEM\_CANO\_AMARELO = pygame.transform.scale2x(pygame.image.load(os.path.join('imgs', 'Sand\_pipe.png')))

IMAGEM\_CANO\_ROXO = pygame.transform.scale2x(pygame.image.load(os.path.join('imgs', 'Vanilla\_pipe.png')))

IMAGEM\_CHAO = pygame.transform.scale2x(pygame.image.load(os.path.join('imgs', 'base.png')))

IMAGEM\_BACKGROUND = pygame.transform.scale2x(pygame.image.load(os.path.join('imgs', 'bg.png')))

pygame.font.init()

FONTE\_PONTOS = pygame.font.SysFont('arial', 30)

FONTE\_GAMEOVER = pygame.font.SysFont('arial', 40)

FONTE\_REINICIAR = pygame.font.SysFont('arial', 30)

FONTE\_RECORDE = pygame.font.SysFont('arial', 30)

class Passaro:

IMAGENS\_PASSARO\_1 = [

pygame.transform.scale2x(pygame.image.load(os.path.join('imgs', 'bird1.png'))),

pygame.transform.scale2x(pygame.image.load(os.path.join('imgs', 'bird2.png'))),

pygame.transform.scale2x(pygame.image.load(os.path.join('imgs', 'bird3.png')))

]

IMAGENS\_PASSARO\_2 = [

pygame.transform.scale2x(pygame.image.load(os.path.join('imgs', 'bobesponja1.png'))),

pygame.transform.scale2x(pygame.image.load(os.path.join('imgs', 'bobesponja2.png'))),

pygame.transform.scale2x(pygame.image.load(os.path.join('imgs', 'bobesponja1.png')))

]

IMAGENS\_PASSARO\_3 = [

pygame.transform.scale2x(pygame.image.load(os.path.join('imgs', 'Deteu3.png'))),

pygame.transform.scale2x(pygame.image.load(os.path.join('imgs', 'Deteu2.png'))),

pygame.transform.scale2x(pygame.image.load(os.path.join('imgs', 'Deteu1.png')))

]

# animações da rotação

ROTAÇAO\_MAXIMA = 25

VELOCIDADE\_ROTAÇAO = 20

TEMPO\_ANIMAÇAO = 5

def \_\_init\_\_(self, x, y, imgs=IMAGENS\_PASSARO\_1):

self.x = x

self.y = y

self.angulo = 0

self.velocidade = 0

self.altura = self.y

self.tempo = 0

self.contagem\_imagem = 0

self.set\_imagens(imgs)

def set\_imagens(self, imgs):

self.IMGS = imgs

self.imagem = self.IMGS[0]

def pular(self):

self.velocidade = -10.5

self.tempo = 0

self.altura = self.y

def mover(self):

# calcular o deslocamento

self.tempo += 1

deslocamento = 1.5 \* (self.tempo \*\* 2) + self.velocidade \* self.tempo

# restringir o deslocamento

if deslocamento > 16:

deslocamento = 16

elif deslocamento < 0:

deslocamento -= 2

self.y += deslocamento

# angulo do passaro

if deslocamento < 0 or self.y < (self.altura + 50):

if self.angulo < self.ROTAÇAO\_MAXIMA:

self.angulo = self.ROTAÇAO\_MAXIMA

else:

if self.angulo > -90:

self.angulo -= self.VELOCIDADE\_ROTAÇAO

def desenhar(self, tela):

# definir qual imagem do pássaro vai ser usado

self.contagem\_imagem += 1

if self.contagem\_imagem < self.TEMPO\_ANIMAÇAO:

self.imagem = self.IMGS[0]

elif self.contagem\_imagem < self.TEMPO\_ANIMAÇAO \* 2:

self.imagem = self.IMGS[1]

elif self.contagem\_imagem < self.TEMPO\_ANIMAÇAO \* 3:

self.imagem = self.IMGS[2]

elif self.contagem\_imagem < self.TEMPO\_ANIMAÇAO \* 4:

self.imagem = self.IMGS[1]

elif self.contagem\_imagem >= self.TEMPO\_ANIMAÇAO \* 4 + 1:

self.imagem = self.IMGS[0]

self.contagem\_imagem = 0

# se o pássaro estiver caindo a asa não vai bater

if self.angulo <= -80:

self.imagem = self.IMGS[1]

self.contagem\_imagem = self.TEMPO\_ANIMAÇAO \* 2

# desenhar imagem

imagem\_rotacionada = pygame.transform.rotate(self.imagem, self.angulo)

pos\_centro\_imagem = self.imagem.get\_rect(topleft=(self.x, self.y)).center

retangulo = imagem\_rotacionada.get\_rect(center=pos\_centro\_imagem)

tela.blit(imagem\_rotacionada, retangulo.topleft)

def get\_mask(self):

return pygame.mask.from\_surface(self.imagem)

class Cano:

DISTANCIA = 250

VELOCIDADE = 10

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

self.x = x

self.altura = 0

self.pos\_topo = 0

self.pos\_base = 0

self.CANO\_TOPO = pygame.transform.flip(IMAGEM\_CANO, False, True)

self.CANO\_BASE = IMAGEM\_CANO

self.CANO\_TOPO\_AZUL = pygame.transform.scale2x(pygame.transform.flip(pygame.image.load(os.path.join('imgs', 'Blue\_pipe.png')), False, True))

self.CANO\_BASE\_AZUL = pygame.transform.scale2x(pygame.image.load(os.path.join('imgs', 'Blue\_pipe.png')))

self.CANO\_TOPO\_AMARELO = pygame.transform.scale2x(pygame.transform.flip(pygame.image.load(os.path.join('imgs', 'Sand\_pipe.png')), False, True))

self.CANO\_BASE\_AMARELO = pygame.transform.scale2x(pygame.image.load(os.path.join('imgs', 'Sand\_pipe.png')))

self.CANO\_TOPO\_ROXO = pygame.transform.scale2x(pygame.transform.flip(pygame.image.load(os.path.join('imgs', 'Purple\_pipe.png')), False, True))

self.CANO\_BASE\_ROXO = pygame.transform.scale2x(pygame.image.load(os.path.join('imgs', 'Purple\_pipe.png')))

self.CANO\_TOPO\_VANILLA = pygame.transform.scale2x(pygame.transform.flip(pygame.image.load(os.path.join('imgs', 'Vanilla\_pipe.png')), False, True))

self.CANO\_BASE\_VANILLA = pygame.transform.scale2x(pygame.image.load(os.path.join('imgs', 'Vanilla\_pipe.png')))

self.passou = False

self.definir\_altura()

def desenhar(self, tela, using\_vanilla\_pipe=False, using\_azul\_pipe = False, using\_amarelo\_pipe =False, using\_roxo\_pipe = False):

if using\_vanilla\_pipe:

tela.blit(self.CANO\_TOPO\_VANILLA, (self.x, self.pos\_topo))

tela.blit(self.CANO\_BASE\_VANILLA, (self.x, self.pos\_base))

elif using\_azul\_pipe:

tela.blit(self.CANO\_TOPO\_AZUL, (self.x, self.pos\_topo))

tela.blit(self.CANO\_BASE\_AZUL, (self.x, self.pos\_base))

elif using\_amarelo\_pipe:

tela.blit(self.CANO\_TOPO\_AMARELO, (self.x, self.pos\_topo))

tela.blit(self.CANO\_BASE\_AMARELO, (self.x, self.pos\_base))

elif using\_roxo\_pipe:

tela.blit(self.CANO\_TOPO\_ROXO, (self.x, self.pos\_topo))

tela.blit(self.CANO\_BASE\_ROXO, (self.x, self.pos\_base))

else:

tela.blit(self.CANO\_TOPO, (self.x, self.pos\_topo))

tela.blit(self.CANO\_BASE, (self.x, self.pos\_base))

def definir\_altura(self):

self.altura = random.randrange(50, 450)

self.pos\_topo = self.altura - self.CANO\_TOPO.get\_height()

self.pos\_base = self.altura + self.DISTANCIA

def mover(self):

self.x -= self.VELOCIDADE

def colidir(self, passaro):

passaro\_mask = passaro.get\_mask()

topo\_mask = pygame.mask.from\_surface(self.CANO\_TOPO)

base\_mask = pygame.mask.from\_surface(self.CANO\_BASE)

distancia\_topo = (self.x - passaro.x, self.pos\_topo - round(passaro.y))

distancia\_base = (self.x - passaro.x, self.pos\_base - round(passaro.y))

topo\_ponto = passaro\_mask.overlap(topo\_mask, distancia\_topo)

base\_ponto = passaro\_mask.overlap(base\_mask, distancia\_base)

if base\_ponto or topo\_ponto:

return True

else:

return False

class Chao:

VELOCIDADE = 5

LARGURA = IMAGEM\_CHAO.get\_width()

IMAGEM = IMAGEM\_CHAO

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

self.y = y

self.x1 = 0

self.x2 = self.LARGURA

def mover(self):

self.x1 -= self.VELOCIDADE

self.x2 -= self.VELOCIDADE

if self.x1 + self.LARGURA < 0:

self.x1 = self.x2 + self.LARGURA

if self.x2 + self.LARGURA < 0:

self.x2 = self.x1 + self.LARGURA

def desenhar(self, tela):

tela.blit(self.IMAGEM, (self.x1, self.y))

tela.blit(self.IMAGEM, (self.x2, self.y))

def desenhar\_tela(tela, passaros, canos, chao, pontos, recorde, game\_over\_flag, using\_vanilla\_pipe=False, using\_azul\_pipe = False, using\_amarelo\_pipe =False, using\_roxo\_pipe = False):

tela.blit(IMAGEM\_BACKGROUND, (0, 0))

for passaro in passaros:

passaro.desenhar(tela)

for cano in canos:

cano.desenhar(tela, using\_vanilla\_pipe, using\_azul\_pipe , using\_amarelo\_pipe, using\_roxo\_pipe)

# Verifica se o jogo está em andamento ou se está no Game Over

# Os numeros 255, 255, 255 são responsaveis por deixar o texto escrito na cor branca, mudar esses valores muda a cor do texto

if not game\_over\_flag:

texto\_pontos = FONTE\_PONTOS.render(f'Pontuação: {pontos}', 1, (255, 255, 255))

texto\_recorde = FONTE\_RECORDE.render(f'Recorde: {recorde}', 1, (255, 255, 255))

tela.blit(texto\_pontos, (TELA\_LARGURA - 10 - texto\_pontos.get\_width(), 10))

tela.blit(texto\_recorde, (TELA\_LARGURA - 10 - texto\_recorde.get\_width(), 70))

if game\_over\_flag:

mensagem\_gameover = FONTE\_GAMEOVER.render("Game Over!", 1, (255, 255, 25))

mensagem\_reiniciar = FONTE\_REINICIAR.render("Aperte Enter para Recomeçar", 1, (25, 255, 255))

# Centralizar as mensagens de "Game Over" e "Aperte Enter para Reiniciar"

# Modificar esses valores de divisão e soma/subtração muda a posiçãC:/Users/20.00961-5/Downloads/FlappyBird-Python-main/FlappyBird-Python-main/imgs/bg.pngo da frase na tela

pos\_x\_gameover = TELA\_LARGURA // 2 - mensagem\_gameover.get\_width() // 2

pos\_y\_gameover = TELA\_ALTURA // 2 - mensagem\_gameover.get\_height() // 2

pos\_x\_reiniciar = TELA\_LARGURA // 2 - mensagem\_reiniciar.get\_width() // 2

pos\_y\_reiniciar = TELA\_ALTURA // 2 + mensagem\_gameover.get\_height() + 100

tela.blit(mensagem\_gameover, (pos\_x\_gameover, pos\_y\_gameover))

tela.blit(mensagem\_reiniciar, (pos\_x\_reiniciar, pos\_y\_reiniciar))

# Centralizar as mensagens de "Pontuação" e "Recorde" na tela de Game Over

texto\_pontos = FONTE\_RECORDE.render(f'Pontuação: {pontos}', 1, (255, 255, 255))

texto\_recorde = FONTE\_RECORDE.render(f'Recorde: {recorde}', 1, (255, 255, 255))

pos\_x\_pontos = TELA\_LARGURA // 2 - texto\_pontos.get\_width() // 2

pos\_y\_pontos = TELA\_ALTURA // 2 + mensagem\_gameover.get\_height() // 2 + 10

pos\_x\_recorde = TELA\_LARGURA // 2 - texto\_recorde.get\_width() // 2

pos\_y\_recorde = TELA\_ALTURA // 2 + mensagem\_gameover.get\_height() // 2 + texto\_pontos.get\_height() + 20

tela.blit(texto\_pontos, (pos\_x\_pontos, pos\_y\_pontos))

tela.blit(texto\_recorde, (pos\_x\_recorde, pos\_y\_recorde))

chao.desenhar(tela)

pygame.display.update()

def tela\_selecao\_passaro(tela):

opcao = None

# Carregar a imagem de fundo

imagem\_background = pygame.transform.scale(pygame.image.load(os.path.join('imgs', 'bg.png')), (TELA\_LARGURA, TELA\_ALTURA))

while opcao not in ('1', '2', '3'):

for evento in pygame.event.get():

if evento.type == pygame.QUIT:

pygame.quit()

quit()

if evento.type == pygame.KEYDOWN:

if evento.key == pygame.K\_1:

opcao = '1'

elif evento.key == pygame.K\_2:

opcao = '2'

elif evento.key == pygame.K\_3:

opcao = '3'

# Desenhar o plano de fundo e a mensagem de escolha do personagem

tela.blit(imagem\_background, (0, 0))

texto\_escolha = FONTE\_PONTOS.render("Escolha o Pássaro: 1, 2 ou 3", True, (255, 255, 255))

texto\_aperte\_tecla = FONTE\_RECORDE.render("Aperte alguma das teclas para jogar", True, (255, 255, 255))

tela.blit(texto\_escolha, (TELA\_LARGURA // 2 - texto\_escolha.get\_width() // 2, TELA\_ALTURA // 2 - 30))

tela.blit(texto\_aperte\_tecla, (TELA\_LARGURA // 2 - texto\_aperte\_tecla.get\_width() // 2, TELA\_ALTURA // 2 + 30))

pygame.display.update()

return opcao

def game\_over(pontos, recorde):

while True:

for evento in pygame.event.get():

if evento.type == pygame.QUIT:

pygame.quit()

quit()

if evento.type == pygame.KEYDOWN:

if evento.key == pygame.K\_RETURN: # Reiniciar o jogo

main()

def carregar\_recorde():

if os.path.isfile('recorde.txt'):

with open('recorde.txt', 'r') as arquivo:

return int(arquivo.read())

else:

return 0

def salvar\_recorde(recorde):

with open('recorde.txt', 'w') as arquivo:

arquivo.write(str(recorde))

def main():

pygame.init()

tela = pygame.display.set\_mode((TELA\_LARGURA, TELA\_ALTURA))

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

opcao\_passaro = tela\_selecao\_passaro(tela)

if opcao\_passaro == '1':

passaros = [Passaro(230, 350, imgs=Passaro.IMAGENS\_PASSARO\_1)]

elif opcao\_passaro == '2':

passaros = [Passaro(230, 350, imgs=Passaro.IMAGENS\_PASSARO\_2)]

else :

passaros = [Passaro(230, 350, imgs=Passaro.IMAGENS\_PASSARO\_3)]

chao = Chao(730)

canos = [Cano(700)]

pontos = 0

recorde = carregar\_recorde()

relogio = pygame.time.Clock()

game\_over\_flag = False

using\_vanilla\_pipe = False

using\_azul\_pipe = False

using\_amarelo\_pipe =False

using\_roxo\_pipe = False

rodando = True

while rodando:

relogio.tick(30)

for evento in pygame.event.get():

if evento.type == pygame.QUIT:

rodando = False

pygame.quit()

quit()

if evento.type == pygame.KEYDOWN:

if evento.key == pygame.K\_SPACE:

if not game\_over\_flag:

for passaro in passaros:

passaro.pular()

if evento.key == pygame.K\_RETURN and game\_over\_flag:

game\_over\_flag = False

pontos = 0

using\_vanilla\_pipe = False

using\_azul\_pipe = False

using\_amarelo\_pipe =False

using\_roxo\_pipe = False

passaros = [Passaro(230, 350)]

canos = [Cano(700)]

if not game\_over\_flag:

for passaro in passaros:

passaro.mover()

chao.mover()

adicionar\_cano = False

remover\_canos = []

for cano in canos:

for i, passaro in enumerate(passaros):

if cano.colidir(passaro):

game\_over\_flag = True

if pontos > recorde:

recorde = pontos

salvar\_recorde(recorde)

if not cano.passou and passaro.x > cano.x:

cano.passou = True

adicionar\_cano = True

cano.mover()

if cano.x + cano.CANO\_TOPO.get\_width() < 0:

remover\_canos.append(cano)

if adicionar\_cano:

pontos += 1

if 40>= pontos >= 20 and not using\_vanilla\_pipe and not using\_azul\_pipe and not using\_amarelo\_pipe and not using\_roxo\_pipe:

using\_vanilla\_pipe = True

canos.append(Cano(600))

elif 60>= pontos >= 40 and not using\_azul\_pipe and not using\_amarelo\_pipe and not using\_roxo\_pipe:

using\_vanilla\_pipe = False

using\_azul\_pipe = True

canos.append(Cano(600))

elif 80>= pontos >= 60 and not using\_vanilla\_pipe and not using\_amarelo\_pipe and not using\_roxo\_pipe:

using\_azul\_pipe = False

using\_amarelo\_pipe = True

canos.append(Cano(600))

elif 100>= pontos >= 80 and not using\_vanilla\_pipe and not using\_azul\_pipe and not using\_roxo\_pipe:

using\_amarelo\_pipe = False

using\_roxo\_pipe = True

canos.append(Cano(600))

else:

canos.append(Cano(600))

for cano in remover\_canos:

canos.remove(cano)

for i, passaro in enumerate(passaros):

if (passaro.y + passaro.imagem.get\_height()) > chao.y or passaro.y < 0:

game\_over\_flag = True

if pontos > recorde:

recorde = pontos

salvar\_recorde(recorde)

desenhar\_tela(tela, passaros, canos, chao, pontos, recorde, game\_over\_flag, using\_vanilla\_pipe,using\_azul\_pipe, using\_amarelo\_pipe, using\_roxo\_pipe)

if game\_over\_flag:

game\_over(pontos, recorde)

if \_\_name\_\_ == '\_\_main\_\_':

main()