import pygame, sys, os

pygame.init()

LV\_1 = [

    '                         ',

    '                         ',

    '                         ',

    'XX     XXX        XX     ',

    'XX                       ',

    'XXXX        XX        XX ',

    'XXXX       XX            ',

    'XX     X XXXX      XX    ',

    '       X XXXX      XXX   ',

    '  P  XXX XXXXXX     XXXX ',

    'XXXXXXXX XXXXXX XX XXXX  ']

TAMANHO\_BLOCO = 64

JANELA\_LARGURA = 800

janela\_altura = len(LV\_1) \* TAMANHO\_BLOCO

tela = pygame.display.set\_mode((JANELA\_LARGURA, janela\_altura))

relogio = pygame.time.Clock()

tela.fill(pygame.Color("gray"))

pygame.display.set\_caption("Menu")

# ============================================== Personagem ========================================================

def bg():

    pygame.draw.line(tela,(0,2,200) ,(0,300), (JANELA\_LARGURA, 300))

class personagem(pygame.sprite.Sprite):

    def \_\_init\_\_(self, posx, posy, tipo):

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

        global direita\_x, esquerda\_x

        self.flip = False

        # pulo

        self.pulo = False

        self.vel\_y = 0

        self.no\_ar = False

        #Tiro

        self.tiro\_cooldown = 0

        # geral

        self.vivo = True

        self.velocidade = 10

        self.direcao = 1

        # Mundo cross

        self.rolagem = 0

        #Animação

        self.tipo = tipo

        self.movimento\_status = 0

        self.index = 0

        self.animacao\_timer = pygame.time.get\_ticks()

        self.animacao = []

        Lista\_tipos\_Animacao = ["Parado", "Andando"] # "Pulo" +

        for animation in Lista\_tipos\_Animacao:

            temp\_list = []

            frames = len(os.listdir(f"img/{self.tipo}/{animation}"))

            for i in range(frames):

                image = pygame.image.load(f'img/{tipo}/{animation}/{i}.png').convert\_alpha()

                temp\_list.append(image)

            self.animacao.append(temp\_list)

        self.image = self.animacao[self.movimento\_status][self.index]

        self.rect = pygame.Rect(posx, posy, image.get\_width(), image.get\_height())

    def update(self):

        if self.tiro\_cooldown > 0:

            self.tiro\_cooldown -= 1

    def Animacao(self, mov\_tipo):

        COOLDONW = 150

        self.movimento\_status = mov\_tipo

#

        if pygame.time.get\_ticks() - self.animacao\_timer > COOLDONW: # ARRUMAR ESSE BGL!!

            self.image = self.animacao[self.movimento\_status][self.index % len(self.animacao[self.movimento\_status])]

            self.animacao\_timer = pygame.time.get\_ticks()

            self.index += 1

        if self.index >= len(self.animacao[self.movimento\_status]):

            self.index = 0

    def Atirar(self):

        if self.tiro\_cooldown == 0:

            self.tiro\_cooldown = 20

            tiro = Bala(self.rect.centerx + (0.6 \* self.rect.size[0] \* self.direcao),self.rect.centery, self.direcao )

            Tiro\_Grupo.add(tiro)

    def update\_acao(self, nova\_acao):

        if nova\_acao != self.movimento\_status:

            self.movimento\_status = nova\_acao

            self.index = 0

            self.animacao\_timer = pygame.time.get\_ticks()

    def move(self, direita\_x, esquerda\_x):

        # Variação do x e y

        dx = 0

        dy = 0

        if direita\_x == True:

            dx += self.velocidade

            self.flip = False

            self.direcao = 1

        if esquerda\_x == True:

            dx -= self.velocidade

            self.flip = True

            self.direcao = -1

        if self.pulo == True and self.no\_ar == False:

            self.vel\_y = -11

            self.pulo = False

            self.no\_ar = True

        self.vel\_y += GRAVIDADE

        if self.vel\_y > 10:

            self.vel\_y = 10

        dy += self.vel\_y

        # Chão

        if self.rect.bottom + dy > 300:

            dy = 300 - self.rect.bottom

            self.no\_ar = False

        # and JANELA\_LARGURA > self.rect.x + self.rect.width

        # self.rect.x > 0

        key = pygame.key.get\_pressed()

        if key[pygame.K\_SPACE]:

            pass

        # if self.tipo == 'Jogador':

        #     if self.rect.right > JANELA\_LARGURA - ROLAGEM\_LIMITE or self.rect.left < ROLAGEM\_LIMITE:

        #         tela\_rolagem = -self.direcao \* self.velocidade

        #         self.rect.x -= self.direcao \* self.velocidade

        self.rect.x += dx

        self.rect.y += dy

        # Jogador\_x = self.rect.centerx

        # if Jogador\_x < 200:

        #     Lv.movimento\_mapa = 8

        #     self.velocidade = 0

    def desenhar(self):

        tela.blit(pygame.transform.flip(self.animacao[self.movimento\_status][self.index], self.flip, False), self.rect)

        pygame.draw.rect(tela, (0, 225, 0), self.rect, 2)

# ============================================== Blocos ========================================================

class Blocos(pygame.sprite.Sprite):

    def \_\_init\_\_(self, pos, tamanho):

        super().\_\_init\_\_()

        self.image = pygame.Surface((tamanho, tamanho))

        self.image.fill(pygame.Color("grey"))

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

    def update(self, x\_dir):  # O nome tem que ser "update" pq ele escreve por cima do método update do pygame

        self.rect.x += x\_dir

# ============================================== LV ========================================================

class Lv:

    def \_\_init\_\_(self, lv\_map, superficie):

        self.display\_surface = superficie

        self.mapa(lv\_map)

        self.movimento\_mapa = 0

    def mapa(self, layout):

        self.sprite\_grupo = pygame.sprite.Group()

        for indexy, linha in enumerate(layout):

            for indexx, coll in enumerate(linha):

                x = indexx \* TAMANHO\_BLOCO

                y = indexy \* TAMANHO\_BLOCO

                if coll == 'X':

                    bloco = Blocos((x, y), TAMANHO\_BLOCO)

                    self.sprite\_grupo.add(bloco)

    def iniciar(self):

        self.sprite\_grupo.update(self.movimento\_mapa)

        self.sprite\_grupo.draw(self.display\_surface)

# ============================================== Bala ========================================================

class Bala(pygame.sprite.Sprite):

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

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

        self.velocidade = 10

        self.image = Tiro\_img

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

        self.rect.center = (x, y)

        self.direcao = direcao

    def update(self):

        self.rect.x += self.direcao \* self.velocidade

        # se saiu da tela

        if self.rect.right < 0 or self.rect.left > JANELA\_LARGURA:

            self.kill()

# ============================================== Fim Classes ========================================================

ROLAGEM\_LIMITE = 200

tela\_rolagem = 0

bg\_rolagem = 0

GRAVIDADE = 0.75

#Bala

Tiro\_Grupo = pygame.sprite.Group()

#IMGS LOAD

Tiro\_img = pygame.image.load("img\gosma.png")

Inimigo1 = personagem(50, janela\_altura - 90, "Rato")

lv = Lv(LV\_1, tela)

blocos\_teste = pygame.sprite.Group(Blocos((10, 10), 200))

Jogador = personagem(50, janela\_altura - 90, "Jogador")

# ============================================== Menu ========================================================

def menu():

    menu\_texto = pygame.image.load("img//Menu.png").convert\_alpha()

    botao\_img = pygame.image.load("img//Botao.png").convert\_alpha()  # image botão

    botao\_img\_rect = botao\_img.get\_rect()

    botao\_img\_rect.center = (JANELA\_LARGURA / 2, janela\_altura / 2)

    tela.fill(pygame.Color("gray"))

    run = True

    while run:

        tela.blit(menu\_texto, (JANELA\_LARGURA / 2 - menu\_texto.get\_width() \* 0.5, 0))

        tela.blit(botao\_img, (botao\_img\_rect.x, botao\_img\_rect.y))

        pygame.draw.rect(tela, (1, 100, 1), botao\_img\_rect, 1)

        mods = pygame.key.get\_mods()

        pos = pygame.mouse.get\_pos()

        for event in pygame.event.get():

            if event.type == pygame.QUIT or mods & pygame.KMOD\_CTRL and event.key == pygame.K\_w:

                sys.exit()

        if botao\_img\_rect.collidepoint(pos):

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

                jogo()

        pygame.display.update()

# ============================================== Jogo ========================================================

def jogo():

    pygame.display.set\_caption("Jogo")

    run = True

    direita\_x = False

    esquerda\_x = False

    Tiro  = False

    while run:

        relogio.tick(60)

        tela.fill(pygame.Color("white"))

        bg()

        # Mapa

        lv.iniciar()

        Tiro\_Grupo.update()

        Tiro\_Grupo.draw(tela)

        # Inimigos

        Jogador.desenhar()

        Inimigo1.desenhar()

        Jogador.update()

        # Pega teclas especiais (Crtl, Alt...)

        key = pygame.key.get\_pressed()

        mods = pygame.key.get\_mods()

        for event in pygame.event.get():

            # Exit

            if event.type == pygame.QUIT or mods & pygame.KMOD\_CTRL and event.key == pygame.K\_w:

                sys.exit()

            # Controles

            if event.type == pygame.KEYDOWN:

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

                    direita\_x = True

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

                    esquerda\_x = True

            if event.type == pygame.KEYUP:

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

                    direita\_x = False

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

                    esquerda\_x = False

            if key[pygame.K\_SPACE]:

                Jogador.pulo = True

            if event.type == pygame.MOUSEBUTTONDOWN:

                Tiro = True

            if event.type == pygame.MOUSEBUTTONUP:

                Tiro = False

        if key[pygame.K\_ESCAPE]:

            menu()

        # Ac~es verificação

        if Jogador.vivo:

            if Tiro:

                Jogador.Atirar()

            if Jogador.no\_ar:

                Jogador.Animacao(1)

            elif direita\_x or esquerda\_x:

                Jogador.Animacao(1)

            else:

                Jogador.Animacao(0)

        Jogador.move(direita\_x, esquerda\_x)

        pygame.display.update()

jogo()

#menu()

# Fontes:

# https://www.youtube.com/watch?v=RRg-3f6tbjY

# https://www.youtube.com/watch?v=YWN8GcmJ-jA&list=PL8ui5HK3oSiGXM2Pc2DahNu1xXBf7WQh-&index=2

# https://github.com/russs123/Shooter/blob/main/shooter\_tut11.py