```
import os
import pygame
from util_imagem import carregar_imagem


""" Entidade do jogador e sua física simples """


class Jogador(pygame.sprite.Sprite):
    def __init__(self, posicao_inicial: tuple[int, int], dir_arquivos: str):
        super().__init__()
        caminho_mario = os.path.join(dir_arquivos, "mario.gif")
        self.imagem = carregar_imagem(caminho_mario, altura_escala=64)
        self.retangulo = self.imagem.get_rect(topleft=posicao_inicial)
        self.velocidade_x = 0.0
        self.velocidade_y = 0.0
        self.no_chao = False

        self.aceleracao_chao = 0.8
        self.aceleracao_ar = 0.5
        self.atrito = 0.85
        self.gravidade = 0.9
        self.forca_pulo = -17
        self.velocidade_maxima_x = 8
        self.velocidade_maxima_queda = 22

    def ler_entrada(self):
        teclas = pygame.key.get_pressed()
        desejado = 0
        if teclas[pygame.K_LEFT] or teclas[pygame.K_a]:
            desejado -= 1
        if teclas[pygame.K_RIGHT] or teclas[pygame.K_d]:
            desejado += 1
        aceleracao = self.aceleracao_chao if self.no_chao else self.aceleracao_ar
        self.velocidade_x += desejado * aceleracao
        if desejado == 0 and self.no_chao:
            self.velocidade_x *= self.atrito
        if self.velocidade_x > self.velocidade_maxima_x:
            self.velocidade_x = self.velocidade_maxima_x
        if self.velocidade_x < -self.velocidade_maxima_x:
            self.velocidade_x = -self.velocidade_maxima_x

    def tentar_pular(self):
        teclas = pygame.key.get_pressed()
        if (teclas[pygame.K_SPACE] or teclas[pygame.K_UP] or teclas[pygame.K_w]) and
self.no_chao:
            self.velocidade_y = self.forca_pulo
            self.no_chao = False
```

```
def aplicar_gravidade(self):
    self.velocidade_y += self.gravidade
    if self.velocidade_y > self.velocidade_maxima_queda:
        self.velocidade_y = self.velocidade_maxima_queda

def mover_e_colidir(self, solidos: list[pygame.Rect]):
    self.retangulo.x += int(self.velocidade_x)
    for bloco in solidos:
        if self.retangulo.colliderect(bloco):
            # indo para a direita
            if self.velocidade_x > 0:
                self.retangulo.right = bloco.left
            # indo para a esquerda
            elif self.velocidade_x < 0:
                self.retangulo.left = bloco.right
            self.velocidade_x = 0
    self.retangulo.y += int(self.velocidade_y)
    self.no_chao = False
    for bloco in solidos:
        if self.retangulo.colliderect(bloco):
            # caindo
            if self.velocidade_y > 0:
                self.retangulo.bottom = bloco.top
                self.velocidade_y = 0
                self.no_chao = True
            # subindo
            elif self.velocidade_y < 0:
                self.retangulo.top = bloco.bottom
                self.velocidade_y = 0

def atualizar(self, solidos: list[pygame.Rect]):
    self.ler_entrada()
    self.tentar_pular()
    self.aplicar_gravidade()
    self.mover_e_colidir(solidos)
```