```
from random import randint, choice

import pygame

from sys import exit

pygame.init()


# Basic setup


Height = 732

Width = 412

screen = pygame.display.set_mode((Height, Width))

pygame.display.set_caption("Ghost Runner")

Clock = pygame.time.Clock()

FPS = 60


game_active=False


#Fonts

text1=pygame.font.Font('Fonts/DragonHunter-9Ynxj.otf', 40)

# hello=text1.render('hello', False, 'red')

# hello_rect=hello.get_rect(midbottom=(366,80))


#Score

start_time=0

score=0

def Score():

    current_T=int(pygame.time.get_ticks()/100)-start_time

    score_surf=text1.render(f'{current_T}',False,(64,64,64))

    score_rect=score_surf.get_rect(center=(366,80))

    screen.blit(score_surf,score_rect)

    return current_T
```

```
class player(pygame.sprite.Sprite):
    def __init__(self):
        super().__init__()
        self.gravty = 0
        self.player_walk = [pygame.image.load('Sprites/player_1.png').convert_alpha(),
                            pygame.image.load(
            'Sprites/player_2.png').convert_alpha(),
            pygame.image.load('Sprites/player_3.png').convert_alpha(), ]
        self.image_index = 0
        self.image = self.player_walk[self.image_index]
        self.player_jump = pygame.image.load(
            'Sprites/player_2.png').convert_alpha()
        self.rect = self.image.get_rect(bottomleft=(81, 370))
        self.mask = pygame.mask.from_surface(self.image)

    def player_input(self):
        key = pygame.key.get_pressed()
        if key[pygame.K_UP] and self.rect.bottom == 370:
            self.gravty = -20

    def gravity(self):
        self.gravty += 1
        self.rect.bottom += self.gravty
        if self.rect.bottom >= 370:
            self.rect.bottom = 370

    def animation(self):
        if self.rect.bottom < 370:
            self.image = self.player_jump
        else:
            self.image_index += 0.1
            if self.image_index >= 3:
                self.image_index = 0
            self.image = self.player_walk[int(self.image_index)]

    def update(self):
        self.player_input()
        self.gravity()
        self.animation()



position = 0
background1 = pygame.image.load(
    'Sprites/Starry night Image.png').convert_alpha()
pygame.transform.scale(background1, (732, 412))



def background():
    global position
    for i in range(0, 2):
        screen.blit(background1, (background1.get_width()*i+position, 0))
        position -= 1
        if abs(position) > background1.get_width():
            position = 0



class obstacle(pygame.sprite.Sprite):
    def __init__(self, type):
        super().__init__()
        if type == 'fly':
            self.frams = [pygame.image.load('Sprites/ghost_1(40x36).png').convert_alpha(),
              pygame.image.load(
                'Sprites/ghost_2.png').convert_alpha(),
              pygame.image.load(
                'Sprites/ghost_3.png').convert_alpha(),
              pygame.image.load(
                'Sprites/ghost_4.png').convert_alpha(),
              pygame.image.load(
                'Sprites/ghost_5.png').convert_alpha(),
              pygame.image.load('Sprites/ghost_6.png').convert_alpha()]
      self.x_pos = 299
    else:
      self.frams = [pygame.image.load('Sprites/skelton_1(35x55).png').convert_alpha(),
              pygame.image.load('Sprites/skelton_2(35x55).png').convert_alpha()]
      self.x_pos = 370
    self.animation_index = 0
    self.image = self.frams[self.animation_index]
    self.rect = self.image.get_rect(
      bottomleft=(randint(900, 1000), self.x_pos))
    self.mask = pygame.mask.from_surface(self.image)

  def animation(self):
    self.animation_index += 0.1
    if int(self.animation_index) >= len(self.frams):
      self.animation_index = 0
    self.image = self.frams[int(self.animation_index)]

  def update(self):
    self.animation()
    self.rect.x -= 3

  def destroy(self):
        if self.rect.x <= -100:
            self.kill()


def collision():
    if pygame.sprite.spritecollide(playr.sprite, obstacle_group, False, pygame.sprite.collide_mask):
        obstacle_group.empty()
        return False
    else:
        return True


playr = pygame.sprite.GroupSingle()
playr.add(player())
obstacle_group = pygame.sprite.Group()


obstacle_timer = pygame.USEREVENT+1
pygame.time.set_timer(obstacle_timer, 900)
while True:
    for event in pygame.event.get():
        if event.type == pygame.QUIT or (event.type == pygame.KEYDOWN and event.key ==
pygame.K_ESCAPE):
            pygame.quit()
            exit()
        if event.type == obstacle_timer and game_active:
            obstacle_group.add(
                obstacle(choice(['fly', 'skelton', 'skelton', 'skelton'])))
        else:
            obstacle_group.empty()
            if event.type==pygame.KEYDOWN and (event.key==pygame.K_UP and game_active==False):
                game_active=True
                start_time=int(pygame.time.get_ticks()/100)
```

```
if game_active:
    background()
    playr.draw(screen)
    obstacle_group.draw(screen)
    obstacle_group.update()
    playr.update()
    score=Score()
    game_active=collision()

else:
    back_image=pygame.image.load('Sprites/background(inactive)732x412.png')
    screen.blit(back_image, (0,0))
    text_massage=text1.render(f'your Score: {score}',False,(111,196,169))
    text_rect=text_massage.get_rect(midbottom=(366,80))

pygame.display.update()
Clock.tick(FPS)
```