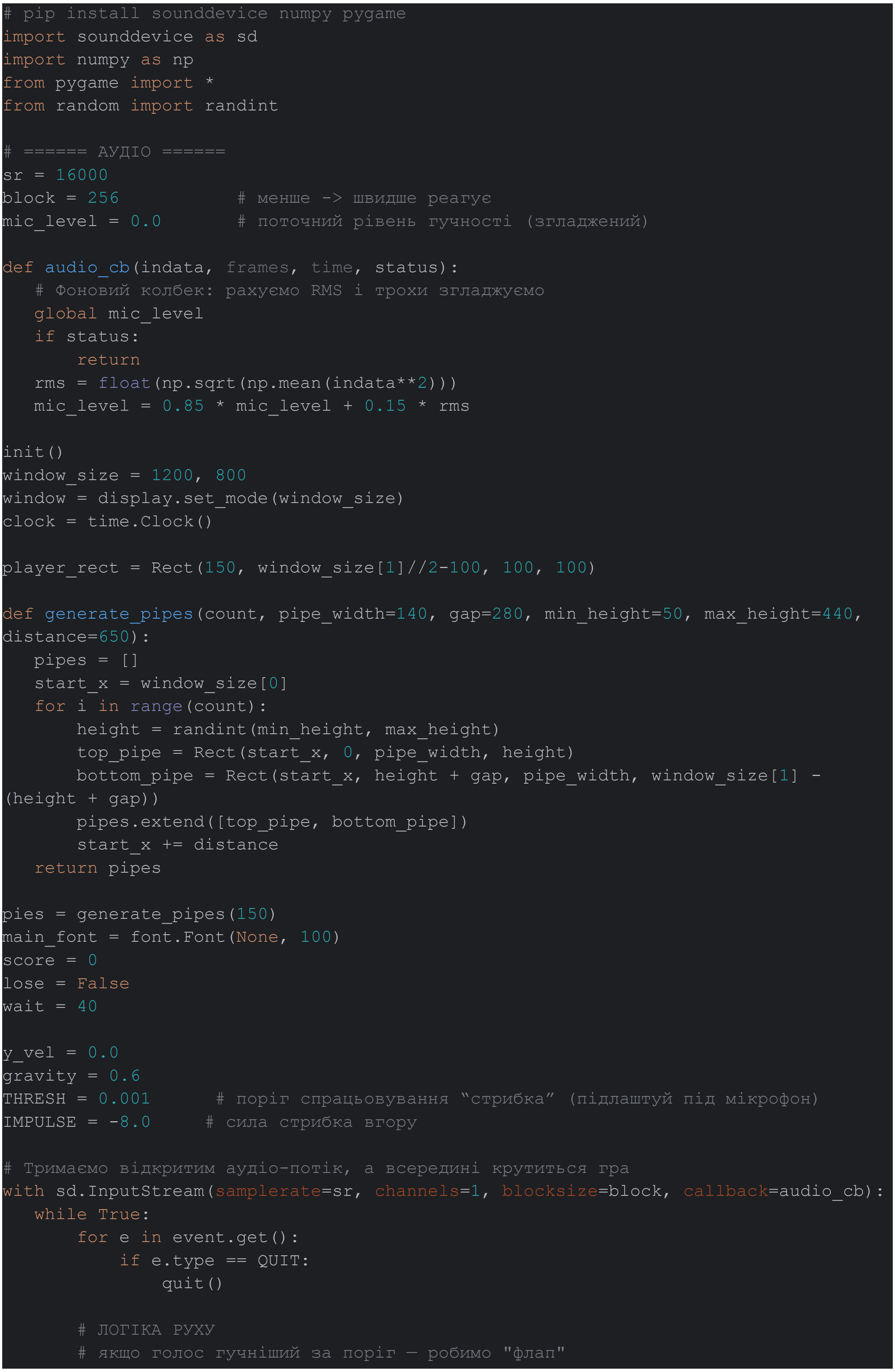
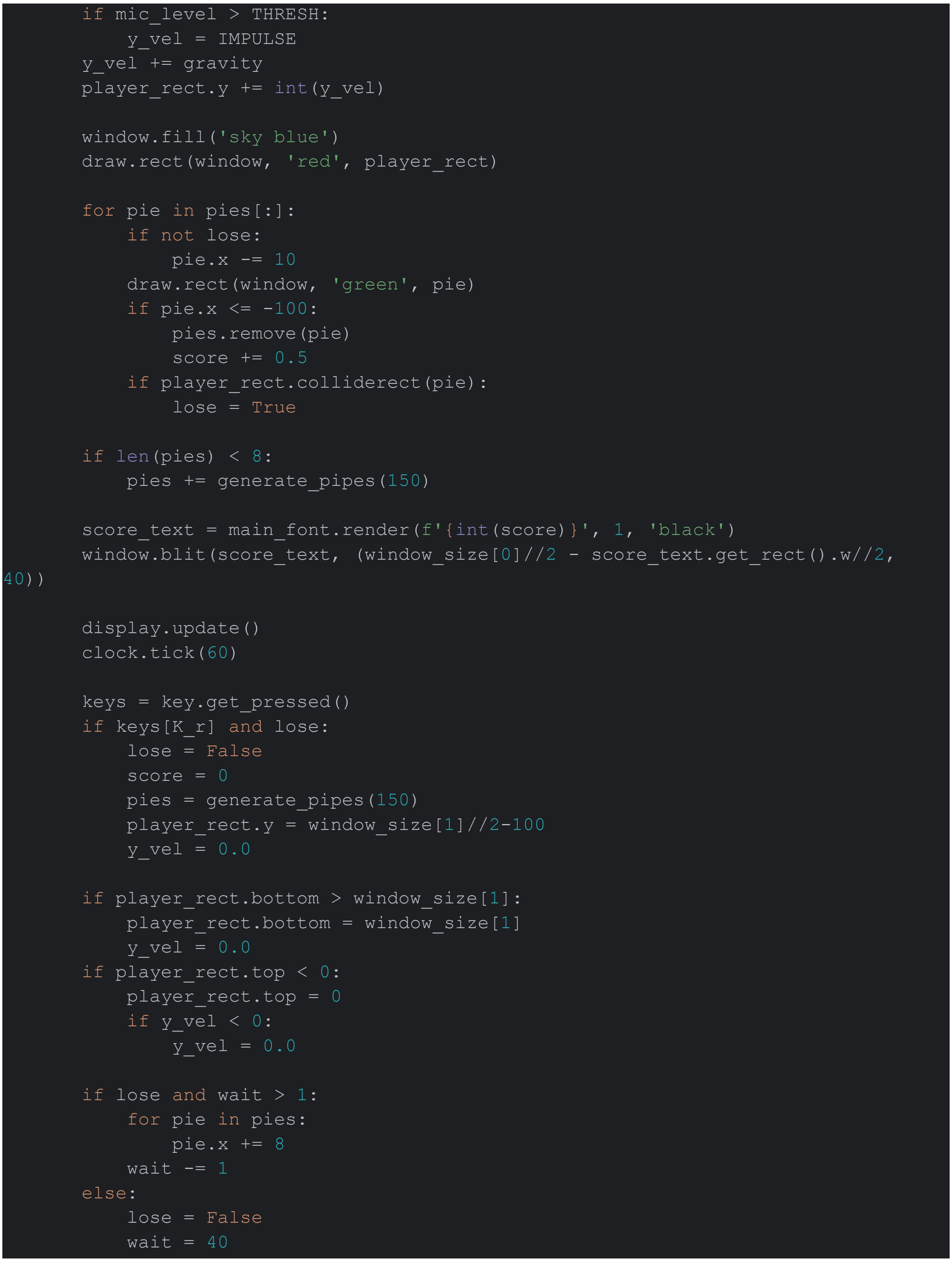


```
# pip install sounddevice numpy pygame
import sounddevice as sd
import numpy as np
from pygame import *
from random import randint

# ====== АУДІО ======
sr = 16000
block = 256           # менше -> швидше реагує
mic_level = 0.0       # поточний рівень гучності (згладжений)

def audio_cb(indata, frames, time, status):
    # Фоновий колбек: рахуємо RMS і трохи згладжуємо
    global mic_level
    if status:
        return
    rms = float(np.sqrt(np.mean(indata**2)))
    mic_level = 0.85 * mic_level + 0.15 * rms

init()
window_size = 1200, 800
window = display.set_mode(window_size)
clock = time.Clock()

player_rect = Rect(150, window_size[1]//2-100, 100, 100)

def generate_pipes(count, pipe_width=140, gap=280, min_height=50, max_height=440,
distance=650):
    pipes = []
    start_x = window_size[0]
    for i in range(count):
        height = randint(min_height, max_height)
        top_pipe = Rect(start_x, 0, pipe_width, height)
        bottom_pipe = Rect(start_x, height + gap, pipe_width, window_size[1] -
(height + gap))
        pipes.extend([top_pipe, bottom_pipe])
        start_x += distance
    return pipes

pies = generate_pipes(150)
main_font = font.Font(None, 100)
score = 0
lose = False
wait = 40

y_vel = 0.0
gravity = 0.6
THRESH = 0.001      # поріг спрацьовування “стрибка” (підлаштуй під мікрофон)
IMPULSE = -8.0     # сила стрибка вгору

# Тримаємо відкритим аудіо-потік, а всередині крутиться гра
with sd.InputStream(samplerate=sr, channels=1, blocksize=block, callback=audio_cb):
    while True:
        for e in event.get():
            if e.type == QUIT:
                quit()

        # ЛОГІКА РУХУ
        # якщо голос гучніший за поріг — робимо "флап"
```



```
        if mic_level > THRESH:
            y_vel = IMPULSE
        y_vel += gravity
        player_rect.y += int(y_vel)

        window.fill('sky blue')
        draw.rect(window, 'red', player_rect)

        for pie in pies[:]:
            if not lose:
                pie.x -= 10
            draw.rect(window, 'green', pie)
            if pie.x <= -100:
                pies.remove(pie)
                score += 0.5
            if player_rect.colliderect(pie):
                lose = True

        if len(pies) < 8:
            pies += generate_pipes(150)

        score_text = main_font.render(f'{int(score)}', 1, 'black')
        window.blit(score_text, (window_size[0]//2 - score_text.get_rect().w//2,
40))

        display.update()
        clock.tick(60)

        keys = key.get_pressed()
        if keys[K_r] and lose:
            lose = False
            score = 0
            pies = generate_pipes(150)
            player_rect.y = window_size[1]//2-100
            y_vel = 0.0

        if player_rect.bottom > window_size[1]:
            player_rect.bottom = window_size[1]
            y_vel = 0.0
        if player_rect.top < 0:
            player_rect.top = 0
            if y_vel < 0:
                y_vel = 0.0

        if lose and wait > 1:
            for pie in pies:
                pie.x += 8
            wait -= 1
        else:
            lose = False
            wait = 40
```