```
import asyncio
import random
import pygame

W, H = 360, 640
SHIP_Y = H - 70

pygame.init()
screen = pygame.display.set_mode((W, H))
pygame.display.set_caption('Space Shooter')
clock = pygame.time.Clock()
font = pygame.font.Font(None, 32)
big = pygame.font.Font(None, 56)

state = {}
stars = [[random.randint(0, W), random.randint(0, H), random.randint(1, 3)]
         for _ in range(50)]


def reset():
    state.update(ship=W / 2, bullets=[], enemies=[], booms=[],
                 fire=0.0, spawn=0.0, score=0, lives=3, over=False)
    state.setdefault('best', 0)


def say(text, f, color, x, y):
    img = f.render(text, True, color)
    screen.blit(img, img.get_rect(center=(x, y)))


def hit(ax, ay, bx, by, r):
    return (ax - bx) ** 2 + (ay - by) ** 2 < r * r


def boom(x, y):
    state['booms'].append({'x': x, 'y': y, 't': 0.0})


def update(dt):
    s = state
    score = s['score']

    # auto fire: more bullets as score grows
    s['fire'] += dt
    if s['fire'] > 0.25:
        s['fire'] = 0
        shots = 1 if score < 20 else 2 if score < 50 else 3
        for i in range(shots):
            offset = (i - (shots - 1) / 2) * 14
            s['bullets'].append({'x': s['ship'] + offset, 'y': SHIP_Y - 22})
```

```
    # new enemies
    s['spawn'] += dt
    if s['spawn'] > max(0.4, 1.0 - 0.015 * score):
        s['spawn'] = 0
        if score >= 5 and random.random() < 0.15:
            s['enemies'].append({'x': random.randint(30, W - 30), 'y': -30,
                                 'hp': 3, 'r': 28, 'pts': 5, 'color': (170, 70, 220)})
        else:
            s['enemies'].append({'x': random.randint(20, W - 20), 'y': -20,
                                 'hp': 1, 'r': 18, 'pts': 1, 'color': (60, 200, 90)})

    # move bullets
    for b in s['bullets'][:]:
        b['y'] -= 600 * dt
        if b['y'] < -20:
            s['bullets'].remove(b)

    # move enemies and check hits
    speed = min(120 + score * 3, 300)
    for e in s['enemies'][:]:
        e['y'] += speed * dt

        for b in s['bullets'][:]:
            if hit(b['x'], b['y'], e['x'], e['y'], e['r'] + 4):
                s['bullets'].remove(b)
                e['hp'] -= 1
                break

        if e['hp'] <= 0:
            s['score'] += e['pts']
            boom(e['x'], e['y'])
            s['enemies'].remove(e)
        elif hit(s['ship'], SHIP_Y, e['x'], e['y'], e['r'] + 14):
            s['lives'] -= 1                    # crashed into the ship
            boom(e['x'], e['y'])
            s['enemies'].remove(e)
        elif e['y'] > H + e['r']:
            s['lives'] -= 1                    # enemy escaped
            s['enemies'].remove(e)

    if s['lives'] <= 0:
        s['over'] = True
        s['best'] = max(s['best'], s['score'])
        s['bullets'] = []
        s['enemies'] = []


def draw():
    screen.fill((10, 10, 40))
    for st in stars:
        pygame.draw.circle(screen, (200, 200, 255), (st[0], int(st[1])), 1)

    s = state
    for b in s['bullets']:
        pygame.draw.rect(screen, (255, 230, 60), (b['x'] - 2, b['y'] - 7, 4, 14))
```

```
    for e in s['enemies']:
        x, y, r = int(e['x']), int(e['y']), e['r']
        pygame.draw.circle(screen, e['color'], (x, y), r)
        for dx in (-r // 3, r // 3):             # two cute eyes
            pygame.draw.circle(screen, (255, 255, 255), (x + dx, y - 3), r // 4)
            pygame.draw.circle(screen, (0, 0, 0), (x + dx, y - 3), r // 8)

    for b in s['booms']:
        radius = int(10 + 40 * b['t'] / 0.3)
        pygame.draw.circle(screen, (255, 140, 0), (int(b['x']), int(b['y'])), radius, 3)

    if not s['over']:
        x = int(s['ship'])
        pygame.draw.polygon(screen, (255, 120, 0),
                            [(x - 7, SHIP_Y + 14), (x + 7, SHIP_Y + 14), (x, SHIP_Y + 30)])
        pygame.draw.polygon(screen, (80, 200, 255),
                            [(x, SHIP_Y - 22), (x - 18, SHIP_Y + 16),
                             (x, SHIP_Y + 8), (x + 18, SHIP_Y + 16)])

    say(f"Score: {s['score']}   Lives: {s['lives']}", font, (255, 255, 255), W // 2, 22)
    if s['over']:
        say('GAME OVER', big, (255, 70, 70), W // 2, H // 2 - 40)
        say(f"Score: {s['score']}   Best: {s['best']}", font, (255, 255, 255), W // 2, H // 2 + 5)
        say('Tap to play again', font, (255, 255, 255), W // 2, H // 2 + 45)


async def main():
    reset()
    running = True
    while running:
        dt = clock.tick(60) / 1000

        # ---------- touch / mouse / keyboard ----------
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                running = False
            elif event.type == pygame.MOUSEBUTTONDOWN:
                if state['over']:
                    reset()
                state['ship'] = event.pos[0]
            elif event.type == pygame.MOUSEMOTION and event.buttons[0]:
                state['ship'] = event.pos[0]
            elif event.type in (pygame.FINGERDOWN, pygame.FINGERMOTION):
                state['ship'] = event.x * W

        keys = pygame.key.get_pressed()
        if keys[pygame.K_LEFT] or keys[pygame.K_a]:
            state['ship'] -= 400 * dt
        if keys[pygame.K_RIGHT] or keys[pygame.K_d]:
            state['ship'] += 400 * dt
        state['ship'] = max(20, min(W - 20, state['ship']))

        # ---------- update ----------
        for st in stars:                         # scrolling stars
```

```
            st[1] = (st[1] + st[2] * 60 * dt) % H
        for b in state['booms'][:]:
            b['t'] += dt
            if b['t'] > 0.3:
                state['booms'].remove(b)
        if not state['over']:
            update(dt)

        draw()
        pygame.display.flip()
        await asyncio.sleep(0)  # needed so the game also runs in the browser

    pygame.quit()


asyncio.run(main())
```