```python
import math
import random


def app_started(app):
    app.timer_delay = 30
    app.pressed_keys = set()
    init(app)


def init(app):
    '''
    Initialize game state. Called when game is started or restarted. Will
    create all necessary variables and set the game state to 'running'.
    '''
    app.game_state = 'running'

    app.spaceship_x = app.width/2
    app.spaceship_y = app.height/2
    app.spaceship_radius = 10
    app.angle = 0
    app.speed_x = 0.3
    app.speed_y = 0.2

    app.bullets = [
        # bullets have the form (x, y, angle), list is initially empty
    ]

    app.stones = [
        # (x, y, radius, speed_x, speed_y)
        (100, 100, 20, 1, 1),
        (200, 300, 30, -1, 1),
        (300, 300, 40, -1, -1),
        (600, 400, 50, 1, -1)
    ]


def key_pressed(app, event):
    app.pressed_keys.add(event.key)
    if app.game_state == 'game_over' and event.key in ['Return', 'Enter']:
        init(app)
    if event.key == 'Space':
        app.bullets.append((app.spaceship_x, app.spaceship_y, app.angle))


def key_released(app, event):
    app.pressed_keys.remove(event.key)


def timer_fired(app):
    move_spaceship(app)
    do_spaceship_steering(app)
    do_move_bullets(app)
    do_move_stones(app)
    check_bullet_hits(app)
    check_stone_collision(app)
    randomly_add_stones(app)


def move_spaceship(app):
    app.spaceship_x += app.speed_x
    app.spaceship_y += app.speed_y

    # Wrap around
    if app.spaceship_x < -app.spaceship_radius:
        app.spaceship_x = app.width + app.spaceship_radius
    elif app.spaceship_x > app.width + app.spaceship_radius:
        app.spaceship_x = -app.spaceship_radius
    if app.spaceship_y < -app.spaceship_radius:
        app.spaceship_y = app.height + app.spaceship_radius
    elif app.spaceship_y > app.height + app.spaceship_radius:
        app.spaceship_y = -app.spaceship_radius


def do_spaceship_steering(app):
    # Rotate spaceship
    if 'Left' in app.pressed_keys:
        app.angle += 0.1
    elif 'Right' in app.pressed_keys:
        app.angle -= 0.1

    # Accelerate spaceship
    elif 'Up' in app.pressed_keys:
        app.speed_x += 0.1 * math.cos(app.angle)
        app.speed_y -= 0.1 * math.sin(app.angle)


def do_move_stones(app):
    for i in range(len(app.stones)):
        x, y, radius, speed_x, speed_y = app.stones[i]
        x += speed_x
        y += speed_y

        # Wrap around
        if x < -radius:
            x = app.width + radius
        elif x > app.width + radius:
            x = -radius

        if y < -radius:
            y = app.height + radius
        elif y > app.height + radius:
            y = -radius

        app.stones[i] = (x, y, radius, speed_x, speed_y)


def do_move_bullets(app):
    # Move bullets
    for i in range(len(app.bullets)):
        x, y, angle = app.bullets[i]
        x += 5 * math.cos(angle)
        y -= 5 * math.sin(angle)
        app.bullets[i] = (x, y, angle)

    # Keep only bullets that are within the screen
    new_bullets = []
    for x, y, angle in app.bullets:
        if 0 <= x <= app.width and 0 <= y <= app.height:
            new_bullets.append((x, y, angle))
    app.bullets = new_bullets


def check_bullet_hits(app):
    '''
    Check if any of the bullets have hit a stone. If so, remove the
    bullet and the stone.
    '''
    for i in range(len(app.bullets)):
        bullet_x, bullet_y, _ = app.bullets[i]
        for j in range(len(app.stones)):
            x, y, radius, _, _ = app.stones[j]
            distance = math.sqrt((x - bullet_x)**2 + (y - bullet_y)**2)
            if distance < radius:
                app.bullets.pop(i)
                app.stones.pop(j)


def check_stone_collision(app):
    for stone in app.stones:
        x, y, radius, _, _ = stone
        dist = math.sqrt((x - app.spaceship_x)**2 + (y - app.spaceship_y)**2)
        if dist < radius + app.spaceship_radius:
            app.game_state = 'game_over'


def randomly_add_stones(app):
    if random.random() < 0.01:
        x = random.randrange(0, app.width)
        y = random.randrange(0, app.height)
        radius = random.randrange(10, 50)
        speed_x = random.uniform(-1, 1)
        speed_y = random.uniform(-1, 1)
        # Move to the edge of the screen according to movement direction
        if random.choice([True, False]):
            x = -radius if speed_x > 0 else app.width + radius
        else:
            y = -radius if speed_y > 0 else app.height + radius

        app.stones.append((x, y, radius, speed_x, speed_y))


def redraw_all(app, canvas):
    canvas.create_rectangle(0, 0, app.width, app.height, fill='black')

    if app.game_state == 'game_over':
        canvas.create_text(
            app.width/2, app.height/2,
            text='Game Over', fill='white', font='Arial 50'
        )
    else:
        draw_spaceship(app, canvas)
        draw_bullets(app, canvas)
        draw_stones(app, canvas)


def draw_spaceship(app, canvas):
    x = app.spaceship_x
    y = app.spaceship_y
    r = app.spaceship_radius
    angle = app.angle

    # Find top in triangle based on angle
    x_tip = x + r * 1.5 * math.cos(angle)
    y_tip = y - r * 1.5 * math.sin(angle)
    x_left = x + r * math.cos(angle + 2/3 * math.pi)
    y_left = y - r * math.sin(angle + 2/3 * math.pi)
    x_right = x + r * math.cos(angle - 2/3 * math.pi)
    y_right = y - r * math.sin(angle - 2/3 * math.pi)

    canvas.create_polygon(
        x_tip, y_tip,
        x_left, y_left,
        x_right, y_right,
        outline='white'
    )

    if 'Up' in app.pressed_keys:
        draw_spaceship_flare(app, canvas, x, y, r, angle)


def draw_spaceship_flare(app, canvas, x, y, r, angle):
    flare_l1_x = x + r * math.cos(angle - 0.9 * math.pi)
    flare_l1_y = y - r * math.sin(angle - 0.9 * math.pi)
    flare_l2_x = x + r * 1.5 * math.cos(angle + 0.9 * math.pi)
    flare_l2_y = y - r * 1.5 * math.sin(angle + 0.9 * math.pi)
    canvas.create_line(
        flare_l1_x, flare_l1_y,
        flare_l2_x, flare_l2_y,
        fill='white'
    )

    flare_r1_x = x + r * math.cos(angle - 0.9 * math.pi)
    flare_r1_y = y - r * math.sin(angle - 0.9 * math.pi)
    flare_r2_x = x + r * 1.5 * math.cos(angle - 0.9 * math.pi)
    flare_r2_y = y + r * 1.5 * math.sin(angle - 0.9 * math.pi)
    canvas.create_line(
        flare_r1_x, flare_r1_y,
        flare_r2_x, flare_r2_y,
        fill='white'
    )


def draw_stones(app, canvas):
    for x, y, r, speed_x, speed_y in app.stones:
        canvas.create_oval(x-r, y-r, x+r, y+r, outline='white')


def draw_bullets(app, canvas):
    for x, y, angle in app.bullets:
        canvas.create_oval(x-2, y-2, x+2, y+2, fill='white', outline='')


if __name__ == '__main__':
    from cmu_112_graphics.event_app import run_app
    run_app(width=800, height=600, title='INF100 Asteroids')
```