Lab 5

Alien Game

In [ ]:
```
alien = Actor('alien')
alien.topright = 0, 10

WIDTH = 500
HEIGHT = alien.height + 20

def draw():
    screen.clear()
    alien.draw()

def update():
    alien.left += 2
    if alien.left > WIDTH:
        alien.right = 0

def set_alien_hurt():
    alien.image = 'alien_hurt'
    sounds.eep.play()

def set_alien_normal():
    alien.image = 'alien'

def set_alien_hurt():
    alien.image = 'alien_hurt'
    sounds.eep.play()
    clock.schedule_unique(set_alien_normal, 1.0)

def on_mouse_down(pos):
    if alien.collidepoint(pos):
        set_alien_hurt()
```

Simple Game

In [ ]:

```
import math
import random
import pygame

arena_width = 800
arena_height = 600

ship_radius = 30

bullet_timer_limit = 0.5
bullet_radius = 5

asteroid_stages = [
    {
        'speed': 120,
        'radius': 15,
    },
    {
        'speed': 70,
        'radius': 30,
    },
    {
        'speed': 50,
        'radius': 50,
    },
    {
        'speed': 20,
        'radius': 80,
    },
]

def reset():
    global ship_x
    global ship_y
    global ship_speed_x
    global ship_speed_y
    global ship_angle
    global bullet_timer
    global bullets
    global asteroids

    ship_x = arena_width / 2
    ship_y = arena_height / 2
    ship_speed_x = 0
    ship_speed_y = 0
    ship_angle = 0

    bullets = []
    bullet_timer = bullet_timer_limit

    asteroids = [
        {
            'x': 100,
            'y': 100,
        },
        {
            'x': arena_width - 100,
```

```
            'y': 100,
        },
        {
            'x': arena_width / 2,
            'y': arena_height - 100,
        }
    ]

    for asteroid in asteroids:
        asteroid['angle'] = random.random() * (2 * math.pi)
        asteroid['stage'] = len(asteroid_stages) - 1

reset()

def update(dt):
    global ship_x
    global ship_y
    global ship_speed_x
    global ship_speed_y
    global ship_angle
    global bullet_timer

    turn_speed = 10

    # take mouse position and calculate ship angle relative to ship position
    mouse_x, mouse_y = pygame.mouse.get_pos()
    ship_angle = math.atan2(mouse_y - ship_y, mouse_x - ship_x)

    # use WASD to position the ship
    if keyboard.D:
        ship_x += 50 * dt

    if keyboard.A:
        ship_x -= 50 * dt

    if keyboard.S:
        ship_y += 50 * dt

    if keyboard.W:
        ship_y -= 50 * dt

    ship_x %= arena_width
    ship_y %= arena_height

    def are_circles_intersecting(a_x, a_y, a_radius, b_x, b_y, b_radius):
        return (a_x - b_x)**2 + (a_y - b_y)**2 <= (a_radius + b_radius)**2

    for bullet in bullets.copy():
        bullet['time_left'] -= dt

        if bullet['time_left'] <= 0:
            bullets.remove(bullet)
            continue

        bullet_speed = 500
        bullet['x'] += math.cos(bullet['angle']) * bullet_speed * dt
        bullet['y'] += math.sin(bullet['angle']) * bullet_speed * dt
```

```
        bullet['x'] %= arena_width
        bullet['y'] %= arena_height

        # die if you shoot yourself
        if are_circles_intersecting(
            ship_x, ship_y, ship_radius,
            bullet['x'], bullet['y'],
            bullet_radius
        ):
            reset()
            break

        for asteroid in asteroids.copy():
            if are_circles_intersecting(
                bullet['x'], bullet['y'], bullet_radius,
                asteroid['x'], asteroid['y'],
                asteroid_stages[asteroid['stage']]['radius']
            ):
                bullets.remove(bullet)

                if asteroid['stage'] > 0:
                    angle1 = random.random() * (2 * math.pi)
                    angle2 = (angle1 - math.pi) % (2 * math.pi)

                    asteroids.append({
                        'x': asteroid['x'],
                        'y': asteroid['y'],
                        'angle': angle1,
                        'stage': asteroid['stage'] - 1
                    })
                    asteroids.append({
                        'x': asteroid['x'],
                        'y': asteroid['y'],
                        'angle': angle2,
                        'stage': asteroid['stage'] - 1
                    })

                asteroids.remove(asteroid)
                break

    bullet_timer += dt

    # if you left click shoot bullet
    if pygame.mouse.get_pressed()[0]:
        if bullet_timer >= bullet_timer_limit:
            bullet_timer = 0

            bullets.append({
                'x': ship_x + math.cos(ship_angle) * ship_radius,
                'y': ship_y + math.sin(ship_angle) * ship_radius,
                'angle': ship_angle,
                'time_left': 4,
            })

    for asteroid in asteroids:
        asteroid_speed = asteroid_stages[asteroid['stage']]['speed']
        asteroid['x'] += math.cos(asteroid['angle']) * asteroid_speed * dt
```

```
        asteroid['y'] += math.sin(asteroid['angle']) * asteroid_speed * dt
        asteroid['x'] %= arena_width
        asteroid['y'] %= arena_height

        if are_circles_intersecting(
            ship_x, ship_y, ship_radius,
            asteroid['x'], asteroid['y'],
            asteroid_stages[asteroid['stage']]['radius']
        ):
            reset()
            break

    if len(asteroids) == 0:
        reset()

def draw():
    screen.fill((0, 0, 0))

    for y in range(-1, 2):
        for x in range(-1, 2):
            offset_x = x * arena_width
            offset_y = y * arena_height

            screen.draw.filled_circle(
                (ship_x + offset_x, ship_y + offset_y),
                ship_radius, color=(0, 0, 255)
            )

            ship_circle_distance = 20
            screen.draw.filled_circle((
                ship_x + offset_x +
                    math.cos(ship_angle) * ship_circle_distance,
                ship_y + offset_y +
                    math.sin(ship_angle) * ship_circle_distance),
                5, color=(0, 255, 255)
            )

            for bullet in bullets:
                screen.draw.filled_circle(
                    (bullet['x'] + offset_x, bullet['y'] + offset_y),
                    bullet_radius, color=(0, 255, 0)
                )

            for asteroid in asteroids:
                screen.draw.filled_circle((
                    asteroid['x'] + offset_x, asteroid['y'] + offset_y),
                    asteroid_stages[asteroid['stage']]['radius'],
                    color=(255, 255, 0)
                )

WIDTH = 800
HEIGHT = 600
```

After playing the game, I had a very hard time winning. So I made the game easier to play by changing the controls to something I am more famliar with.

First, the original game required you to use the left and right arrow keys to aim the ship's gun. The right arrow key, will turn the gun clockwise and the left arrow key would turn counter clockwise. This made it really challenging to aim so I made the program take my mouse position to aim the ship. This was done by using an arctan of the y position of the mouse minus the ship's y position divided by the x poisiton of the mouse minus the ship's x position.

Secondly, the original game was limited by only letting you move the ship using the up arrow key and move forward based on the the ship's direction. To move the ship easier, I just programmed the ship to move with the WASD keys like most PC video games.

Third, the original game required you to use the S key to shoot, but that was being used to move the ship. So I just made the ship shoot by a mouse click. Holding the mouse button will continously shoot.

I also added a feature in the game so that you can die if you accidentally shoot yourself with your own bullet.

All of these modifications can be found in near my comments in the code.

In [ ]: