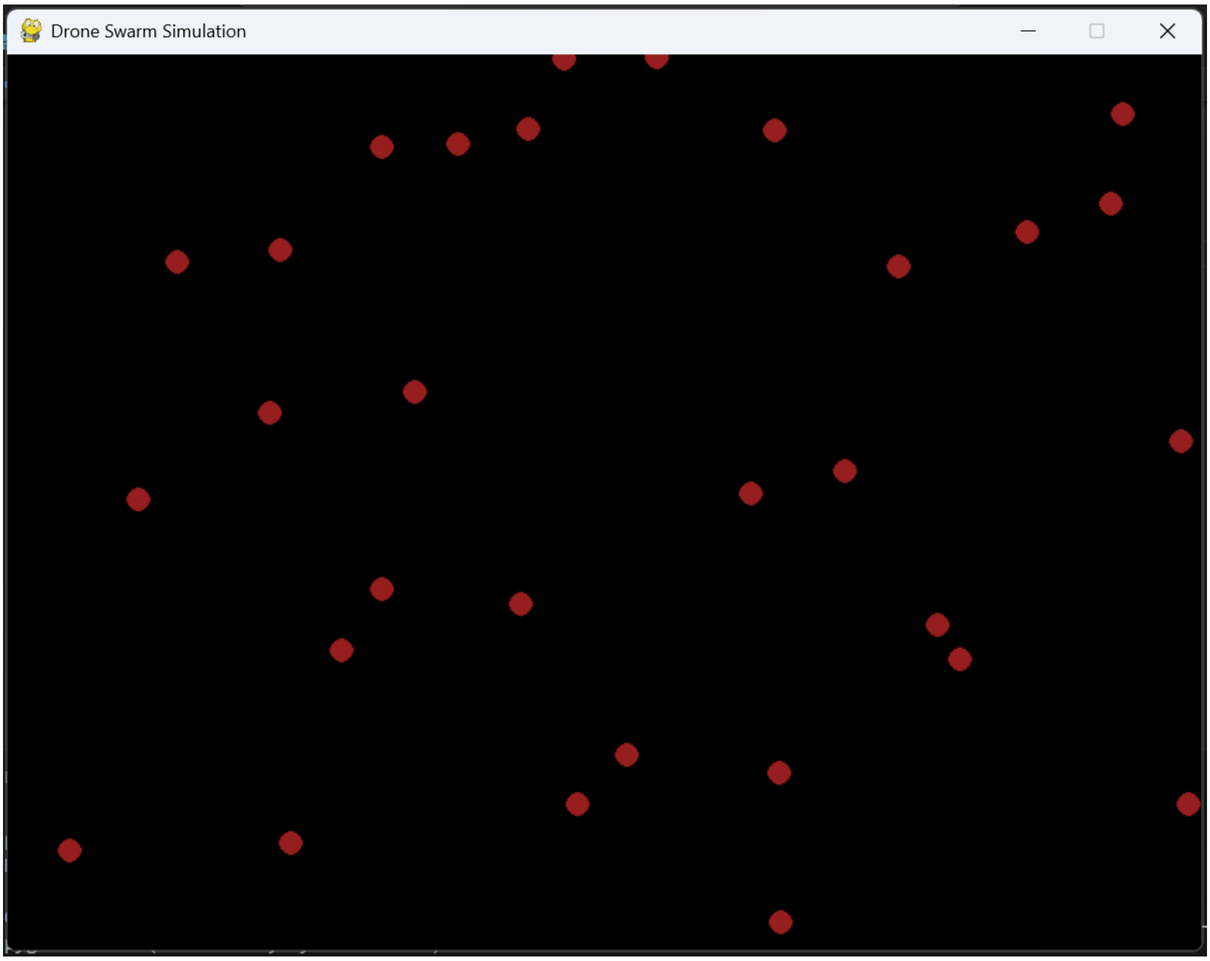
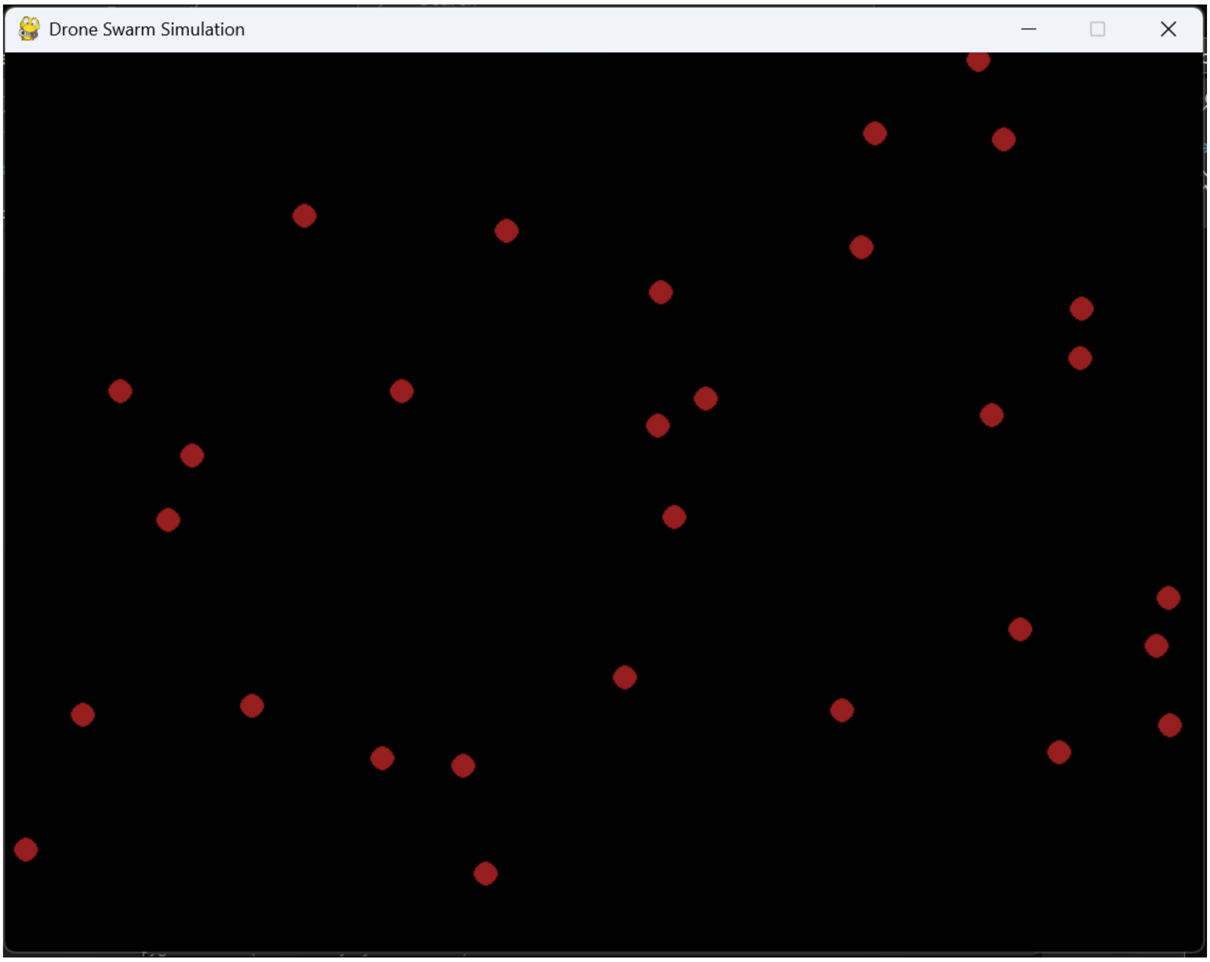
Code:-

```
import pygame

import random

import math


pygame.init()

screen = pygame.display.set_mode((800, 600))

pygame.display.set_caption("Drone Swarm Simulation")


black = (0, 0, 0)

white = (255, 255, 255)


class Drone:

   def __init__(self):

      # it stores the random position of drone

      self.pos = pygame.math.Vector2(random.randint(0, 800), random.randint(0, 600))


      # select random angle to move the drone in any direction

      angle = random.uniform(0, math.pi * 2)


      # velocity vector (at last, 1.7 denotes speed 2pixels/frame)

      self.vel = pygame.math.Vector2(math.cos(angle), math.sin(angle)) * 1.7


      #drone size

      self.size = 8


      # drone colour

      self.color = (150,30,30)


   def update(self, drones):
      # it is a zero vector / initially separation_force will be 0
      separation_force = pygame.math.Vector2()

      # this for loop is for calculating the net velocity based on separation force that is also based on the difference
      # between neighbour and that particular drone itself.
      # it will calcualte the total seperation_force and after getting that net force
      # we will assign net velocity according to that
      for other in drones:
          if other is self:
              continue
          distance = self.pos.distance_to(other.pos)
          if distance < 50 and distance >0:
              diff = self.pos - other.pos
              if diff.length() > 0:
                  diff = diff.normalize() / distance
              separation_force += diff

      self.vel += separation_force

      # speed limit , speed of drone can't exceeds this spped limit
      max_speed = 4
      if self.vel.length() > max_speed:
          self.vel.scale_to_length(max_speed)

      # resultant direction of drone
      self.pos += self.vel

      # if drone go out of the window
      if self.pos.x < 0: self.pos.x = 800
      if self.pos.x > 800: self.pos.x = 0
      if self.pos.y < 0: self.pos.y = 600
      if self.pos.y > 600: self.pos.y = 0

  def draw(self, surface):
      pygame.draw.circle(surface, self.color, (int(self.pos.x), int(self.pos.y)), self.size)

# Create swarm
drones = [Drone() for _ in range(30)]

running = True
clock = pygame.time.Clock()

while running:
  screen.fill(black)

  for event in pygame.event.get():
      if event.type == pygame.QUIT:
          running = False

  for drone in drones:
      drone.update(drones)
      drone.draw(screen)

  # show drones on the screen.
  pygame.display.flip()
  clock.tick(60)

pygame.quit()
```



Drone Swarm Simulation



Drone Swarm Simulation