Система частиц. Применение словаря

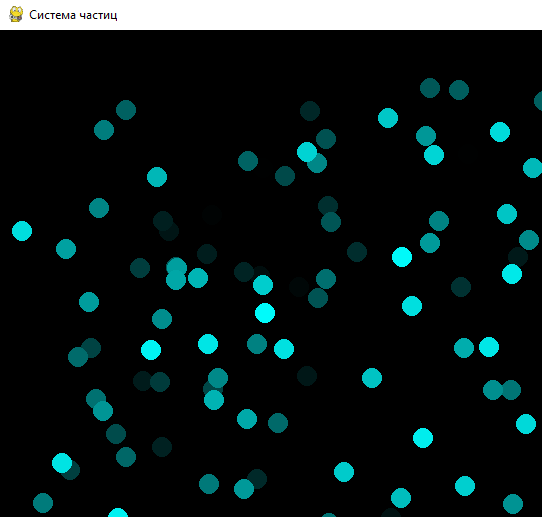
from pygame import \*  
from random import \*  
  
size = W, H = 800, 600  
FPS = 50;  
OK = 1  
S = display.set\_mode(size)  
display.set\_caption('Система частиц')  
clock = time.Clock()  
  
particles = []  
  
  
def update():  
 particle = {  
 'pos': [randrange(W), randrange(H)],  
 'velocity': [uniform(-3, 3), uniform(-3, 3)]  
 }  
 particles.append(particle)  
  
 for p in particles:  
 p['pos'][0] += p['velocity'][0]  
 p['pos'][1] += p['velocity'][1]  
  
  
def render():  
 for p in particles:  
 draw.circle(S, 'green', p['pos'], 10)  
  
  
while OK:  
 for e in event.get():  
 if e.type == KEYUP and e.key == K\_ESCAPE or e.type == QUIT:  
 OK = 0  
  
 render()  
 update()  
 display.update() *# Обновляет часть отображения интерфейса* S.fill('red')  
 clock.tick(FPS)

далее программируем время жизни и исчезание частиц

from pygame import \*  
from random import \*  
  
size = W, H = 800, 600  
FPS = 50;  
OK = 1  
S = display.set\_mode(size)  
display.set\_caption('Система частиц')  
clock = time.Clock()  
  
particles = []  
  
  
def update():  
 particle = {  
 'pos': [randrange(W), randrange(H)],  
 'velocity': [uniform(-3, 3), uniform(-3, 3)],  
 'life': 128  
 }  
 particles.append(particle)  
  
 for p in particles:  
 p['pos'][0] += p['velocity'][0]  
 p['pos'][1] += p['velocity'][1]  
 p['life'] -= 1  
 if p['life'] <= 0:  
 particles.remove(p)  
  
  
def render():  
 for p in particles:  
 k = p['life'] / 128  
 draw.circle(S, (k, 255 \* k, 255 \* k), p['pos'], 10)  
  
  
  
while OK:  
 for e in event.get():  
 if e.type == KEYUP and e.key == K\_ESCAPE or e.type == QUIT:  
 OK = 0  
  
 render()  
 update()  
 display.update() *# Обновляет часть отображения интерфейса* S.fill('black')  
 clock.tick(FPS)

Программируем движение частиц в сторону мышки

def update():  
 for \_ in range(10):  
 particle = {  
 'pos': [randrange(W), randrange(H)],  
 'velocity': [uniform(-3, 3), uniform(-3, 3)],  
 'life': 128  
 }  
 particles.append(particle)  
  
 for p in particles:  
 p['pos'][0] += p['velocity'][0]  
 p['pos'][1] += p['velocity'][1]  
 p['life'] -= 1  
 if p['life'] <= 0:  
 particles.remove(p)  
  
 mp = mouse.get\_pos()  
 dx = mp[0] - p['pos'][0]  
 dy = mp[1] - p['pos'][1]  
 p['velocity'] = (dx, dy)



for p in particles:  
 p['pos'][0] += p['velocity'][0]  
 p['pos'][1] += p['velocity'][1]  
 p['life'] -= 1  
 if p['life'] <= 0:  
 particles.remove(p)  
  
 mp = mouse.get\_pos()  
 dx = mp[0] - p['pos'][0]  
 dy = mp[1] - p['pos'][1]  
 length = sqrt(dx \*\* 2 + dy \*\* 2)  
 dx /= length  
 dy /= length  
 p['velocity'] = (dx, dy)